aval Research Laboratory
ashington, DC 20375-5000

NRL/MR/5120—92-6920

The KRAKEN Normal Mode Program

M. B. PORTER

Applied Ocean Acoustics Branch
Acoustics Division

May 22, 1992

Prepared jointly with SACLANT Undersea Research Centre

92

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 22, 1992	

4. TITLE AND SUBTITLE

The KRAKEN Normal Mode Program

5. FUNDING NUMBERS

PE - 62314N
PR - RJ14C02

6. AUTHOR(S)

M. B. Porter

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/5120—92-6920

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Chief of Naval Research
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Prepared jointly with SACLANT Undersea Research Centre.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In the late 1970's several normal-mode models existed which were widely used for predicting acoustic transmission-loss in the ocean; however, each had its own problems. Typical difficulties included numerical instabilities for certain types of sound-speed profiles and failures to compute a complete set of ocean modes. In short, there was a need for a model that was robust, accurate, and efficient. In order to resolve these problems a new algorithm was developed forming the basis for the KRAKEN normal mode model.

Over subsequent years, KRAKEN was greatly extended, with options for modeling ocean environments that are range-independent, range-dependent or fully 3-dimensional. The current version offers a specialist a vast number of options for treating ocean-acoustics problems (or more generally acousto-elastic waveguides). On the other hand, it is easy for a less sophisticated user to learn the small subset of tools needed for the common problem of transmission-loss modeling in range-independent ocean environments.

This report addresses the need for a more complete user's guide to supplement the on-line help files. The first chapters give a fairly technical description of the mathematical and numerical basis of the model. Additional chapters give a simpler description of its use and installation in a manner that is accessible to less scientifically-oriented readers.

14. SUBJECT TERMS

Normal model
Underwater Sound
Acoustics

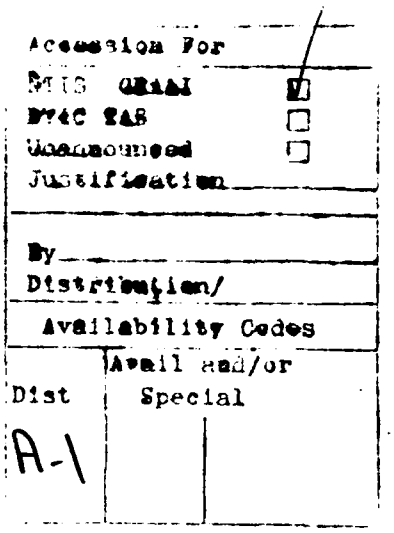
15. NUMBER OF PAGES
99

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std 239-18
298-102



Contents

Chapter 1

Introduction

Normal mode methods have been used for many years in underwater acoustics. One of the earliest papers was published in 1948 by Pekeris [1] who developed the theory for a simple two-layer model (ocean and sediment) with constant sound speed in each layer. Progress in the development of normal-mode methods is represented in an excellent summary given by Williams[2] and published in 1970. Today, there are many models available that are based on normal modes [3–12]. With respect to Pekeris's original work, these models allow for a more detailed description of both the ocean and sediment sound-speed profiles.

Work on KRAKEN[1] was begun in 1980 as part of the author's dissertation with the objective of developing a normal mode model which was more robust, accurate and efficient[13,14]. The basic algorithm was then extended to treat a more sophisticated ocean model in which the *elastic* properties of the ocean bottom are included[15]. At the time, elastic normal mode codes were widely used by seismologists but not very familiar to the ocean-acoustics community. Additional work was done to include the effects of shear flows (e.g. ocean currents) [16].

The KRAKEN model was initially developed as a research code to evaluate new algorithms. As such it required numerous modifications to be usable as a production code. This work was begun at the Naval Ocean Systems Center and continued at the Naval Research Laboratory in support of the research on matched-field processing.

The extension to three-dimensional environments [17] was also done at NRL. That work led to the program FIELD3D which formed the nucleus of the Wide-Area Rapid Acoustic Prediction (WRAP) system. WRAP has been extended by a number of people and now includes options for noise modeling[18] and can include this information to predict array performance in complex 3-D environments with different kinds of signal processing schemes. This report documents only the KRAKEN model, not the complete WRAP system.

When the original KRAKEN work was done the algorithm was incorporated into the very popular SNAP model at SACLANTCEN and subsequently renamed to SUPERSNAP. Since 1984 SUPERSNAP has become the standard and is now simply

[1] In answer to the most frequently asked question: **KRAKEN** krăk-en n [Norw dial.] (1755): a fabulous Scandinavian sea monster. (Webster's Ninth New Collegiate Dictionary)

Manuscript approved November 27, 1991.

Figure 1.1: Cachet engraving of a KRAKEN (from the Canadian Illustrated News, October 27, 1877).

referred to as SNAP. As a result the current version of SNAP and KRAKEN provide *identical* results when run on the same problem. The execution time is also identical.

In essence, the difference between the two models is that KRAKEN provides a large number of extensions and options, whose presence is an advantage to a sophisticated user and a disadvantage to the uninitiated. At SACLANTCEN, both models are being maintained: KRAKEN is recommended for more experienced modelers or for those requiring 3-D capability and SNAP recommended for those interested simply in transmission loss calculations. Amongst the features of KRAKEN are:

- efficient eigenvalue finding techniques guaranteed to converge
- stable eigenfunction calculation even with multiple ducts
- ability to handle multilayered environments
- inclusion of stratified *elastic* layers
- inclusion of interfacial roughness
- tabulated surface and bottom reflection coefficients
- choice of perturbational or exact treatment of loss
- calculation of leaky modes
- free, rigid, and homogenous half-space options for boundary conditions
- adiabatic or coupled mode options for range-dependent problems

- tilted and displaced array calculations
- high-accuracy via extrapolation
- extension to 3-dimensionally varying problems

This report is organized as follows. Chapter 2 provides a fairly technical description of the mathematical basis for normal modes. This material is intended as a tutorial on normal modes and makes limited reference to the specifics of the KRAKEN model. Chapter 3 discusses the numerical treatment of the modal equation and Chap. 4 provides information on running the program. In Chap. 5 we present a number of test problems which exercise different parts of the code. These problems are not particularly physical but they do provide a means of verifying the model on a new installation. In addition, they illustrate the set-up of the input file for different types of environmental scenarios.

Chapter 2

Mathematical Formulation

The normal mode solution involves solving a one-dimensional equation which is very similar to that of a vibrating string. In this analogy, the "frequencies" of vibration give the horizontal wavenumbers associated with the modal propagation and a varying string thickness corresponds to a change in sound speed with depth. The position where the string is plucked corresponds to the source depth and the relative excitation of the ocean modes depends on the source depth just as the harmonic balance of a note is affected by the position where the string is plucked.

We will begin with a simple derivation of the normal mode equations based on separation of variables. This provides a quick means of introducing the gross features of the normal mode approach in Sect. 2.2. In Sect. 2.3, we present a generalized derivation which starts with the spectral representation of the acoustic field. This derivation is necessary for treating the more complex problems involving interfacial roughness, homogeneous halfspaces, or other more complicated boundary conditions which are treated in Sects. 2.5–2.7. Normal modes are normally thought of principally in the context of range-independent problems, however, they can be extended in various ways to both range-dependent problems and fully three-dimensional problems. These extensions are discussed in Sects. 2.8 and 2.9 respectively.

2.1 Derivation

The problem we consider is that of calculating the response to an isotropic point source in a stratified (i.e. range-independent) acoustic medium. The scenario is indicated schematically in Fig. 2.1. Within a layer the solution is governed by the acoustic wave equation:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla P \right) - \frac{1}{\rho c^2(z)} P_{tt} = -s(t) \frac{\delta(z - z_s)\delta(r)}{2\pi r}, \tag{2.1}$$

where $P(r, z, t)$ is the acoustic pressure as a function of depth z, range r, and time t. In addition, $s(t)$ is the isotropic point source, $\rho(z)$ is the density and $c(z)$ is the sound speed. For the moment we assume that the surface is a pressure release boundary and that at some sufficiently great depth D, the boundary can be treated

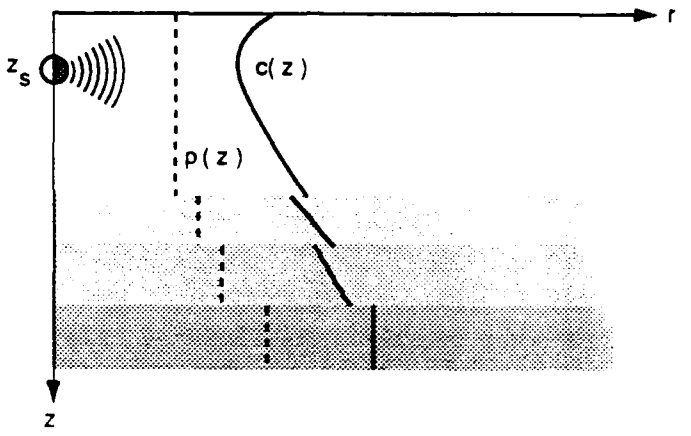


Figure 2.1: Schematic of the range-independent environmental scenario.

as perfectly rigid:

$$\begin{aligned} P(r,0,t) &= 0, \\ P_z(r,D,t) &= 0. \end{aligned} \tag{2.2}$$

(The perfectly rigid boundary is often used as a first approximation to the ocean bottom, but in fact a pressure-release bottom is almost always a better approximation: long range propagation is dominated by rays at grazing incidence where even a relatively fast bottom acts like a pressure release boundary.) Furthermore, we require

$$P(r,z,t) \quad \text{outgoing as } r \to \infty.$$

When discontinuous interfaces are present the wave equation applies within each smooth layer and interface conditions requiring continuity of pressure and normal displacement are imposed.

The first simplification we make is to assume that we are only interested in a single frequency component, that is, the response of the ocean to a continuous hum. Thus, we assume that the source time series has the form

$$s(t) = e^{-i\omega t}, \tag{2.3}$$

which leads to a pressure field with the same harmonic time-dependence [1]:

$$P(r,z,t) = p(r,z)e^{-i\omega t}. \tag{2.4}$$

Making this substitution in the wave equation yields the so-called Helmholtz equation or reduced wave-equation:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial p}{\partial r}\right) - \rho(z)\frac{\partial}{\partial z}\left(\frac{1}{\rho(z)}\frac{\partial p}{\partial z}\right) - \frac{\omega^2}{c^2(z)}p = \frac{-\delta(z-z_s)\delta(r)}{2\pi r}. \tag{2.5}$$

[1] This choice of $e^{-i\omega t}$ is probably the most common: KRAKEN is actually implemented using the opposite convention to be compatible with certain FFP models. You need to be alert to this point if you are using model output to do matched-field processing.

Using the technique of separation of variables, we seek a solution of the unforced equation (with the source removed) in the form $p(r,z) = Z(z)R(r)$. Thus we find,

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial R}{\partial r}\right)\right] + \left[\rho(z)\frac{\partial}{\partial z}\left(\frac{1}{\rho(z)}\frac{\partial Z}{\partial z}\right) + \frac{\omega^2}{c^2(z)}\right] = 0. \tag{2.6}$$

The two components in square brackets are functions of r and z respectively. Thus, the only way the equation can be satisfied is if each component is equal to a constant. Denoting this separation constant by k^2 we obtain the modal equation:

$$\begin{aligned} \rho(z)\frac{d}{dz}\left(\frac{1}{\rho(z)}\frac{dZ(z)}{dz}\right) + \left(\frac{\omega^2}{c^2(z)} - k^2\right)Z(z) &= 0, \\ Z(0) &= 0, \\ \frac{dZ}{dz}(D) &= 0. \end{aligned} \tag{2.7}$$

The modal equation is a classical Sturm-Liouville eigenvalue problem whose properties are well-known [28]. (We assume for the moment that $\rho(z)$ and $c(z)$ are real functions.) A brief summary of these properties follows. The modal equation has an infinite number of solutions (modes) which are characterized by a mode shape function $Z_m(z)$ and a horizontal propagation constant k_m. These horizontal propagation constants, are all distinct. The function $Z_m(z)$ is an eigenfunction and k_m or k_m^2 is an eigenvalue. The mth mode has m zeroes in the interval $[0, D]$ and the corresponding eigenvalues, k_m^2 are all real and we choose to order them such that $k_1^2 > k_2^2 > \cdots$. One can also show that all the eigenvalues are less than ω/c_{min} where c_{min} is the lowest sound speed in the problem.

In addition, the modes of such Sturm-Liouville problems are *orthogonal*. That is,

$$\int_0^D \frac{Z_m(z)Z_l(z)}{\rho(z)}\,dz = 0. \qquad \text{for } m \neq n. \tag{2.8}$$

The solutions of the modal equation are arbitrary to a multiplicative constant as is easily seen from Eq. (2.7). In order to simplify certain results, we shall assume that the modes are scaled (normalized) so that

$$\int_0^D \frac{Z_m^2(z)}{\rho(z)}\,dz = 1. \tag{2.9}$$

With this scaling, the modes form an ortho*normal* set. Furthermore, the set is *complete* which means we can represent an arbitrary function as a sum of the normal modes. Thus, we write the pressure as

$$p(r,z) = \sum_{m=1}^{\infty} R_m(r)Z_m(z). \tag{2.10}$$

If we now substitute this into Eq. (2.5) we obtain:

$$\sum_{m=1}^{\infty}\left\{\frac{\rho(z)}{r}\frac{\partial}{\partial r}\left(\frac{r}{\rho(z)}\frac{\partial R_m(r)}{\partial r}\right)Z_m(z)\right.$$

$$+R_m(r)\left[\rho(z)\frac{d}{dz}\left(\frac{1}{\rho(z)}\frac{dZ(z)}{dz}\right)+\frac{\omega^2}{c^2(z)}Z_m(z)\right]\Bigg\} = \frac{-\delta(r)\delta(z-z_s)}{2\pi r}.$$

This implies,

$$\sum_{m=1}^{\infty}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial R_m(r)}{\partial r}\right)Z_m(z)+k_m^2R_m(r)Z_m(z)\right] = \frac{-\delta(r)\delta(z-z_s)}{2\pi r}. \tag{2.11}$$

Next we apply the operator:

$$\int_0^D(\cdot)\frac{Z_l(z)}{\rho(z)}dz, \tag{2.12}$$

to this equation. Because of the orthogonality property (Eq. 2.8) only the nth term in the sum remains. This gives us:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial R_l(r)}{\partial r}\right)+k_l^2R_l(r) = \frac{-\delta(r)Z_l(z_s)}{2\pi r\rho(z_s)}. \tag{2.13}$$

This is a standard equation whose solution is given in terms of a Hankel function[28] as

$$R_l(r) = \frac{i}{4\rho(z_s)}Z_l(z_s)H_0^{(1,2)}(k_lr). \tag{2.14}$$

The choice of $H_0^{(1)}$ or $H_0^{(2)}$ is determined by the radiation condition (that energy should be radiating outward as $r\rightarrow\infty$). Since we have suppressed a time-dependence of the form $e^{-i\omega t}$ with ω positive and since the k_l are chosen to lie in the right-half plane we shall take the Hankel function of the first kind. Putting this all together, one finds that,

$$p(r,z) = \frac{i}{4\rho(z_s)}\sum_{m=1}^{\infty}Z_m(z_s)Z_m(z)H_0^{(1)}(k_mr), \tag{2.15}$$

or, using the asymptotic approximation to the Hankel function.

$$p(r,z)\approx\frac{i}{\rho(z_s)\sqrt{8\pi r}}e^{-i\pi/4}\sum_{m=1}^{\infty}Z_m(z_s)Z_m(z)\frac{e^{ik_mr}}{\sqrt{k_m}}. \tag{2.16}$$

We normally plot not the complex pressure field but transmission loss. Transmission loss is defined by:

$$TL(r,z) = -20\log\frac{p(r,z)}{p^0(r=1)}, \tag{2.17}$$

where

$$p^0(r) = \frac{e^{ik_0r}}{4\pi r} \tag{2.18}$$

is the pressure for the source in free space. Thus one may write:

$$TL(r,z)\approx -20\log\frac{1}{\rho(z_s)}\sqrt{\frac{2\pi}{r}}\sum_{m=1}^{\infty}Z_m(z_s)Z_m(z)\frac{e^{ik_mr}}{\sqrt{k_m}}. \tag{2.19}$$

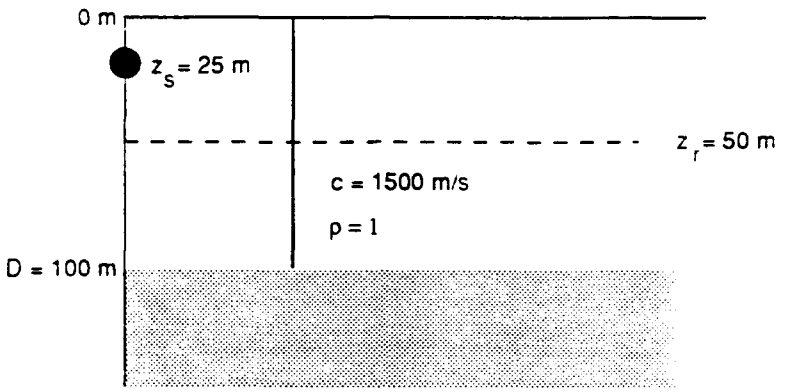


Figure 2.2: Schematic of the isovelocity problem.

In some cases it is useful to calculate an incoherent transmission loss defined by:

$$TL^{Inc}(r,z) \approx -20\log\frac{1}{\rho(z_s)}\sqrt{\frac{2\pi}{r}}\sqrt{\sum_{m=1}^{\infty}\left|\frac{Z_m(z_s)Z_m(z)}{\sqrt{k_m}}\right|}. \tag{2.20}$$

If one is comparing to measured data which has been taken by averaging over frequency one can often simulate the resulting smoothed result by an incoherent TL calculation. Incoherent transmission loss is also often appropriate for shallow water problems. In shallow water the modes are normally bottom-interacting and bottom properties are usually poorly known. This in turn means that the detailed interference pattern predicted by a coherent TL calculation is not physically meaningful.

2.2 The Isovelocity Problem

The principal numerical problem is to solve for the normal modes $Z_m(z)$ corresponding to Eq. (2.7). The sound speed profile, $c(z)$ assumes a fairly arbitrary form so simple analytical techniques are generally not useful. On the other hand, it is instructive to consider some simple profiles in order to understand the qualitative features of modal problems. The simplest such case is the *isovelocity* profile with unit density as shown in Fig. 2.2. The general solution is

$$Z_m(z) = A\sin\gamma z + B\cos\gamma z, \tag{2.21}$$

where

$$\gamma = \sqrt{\frac{\omega^2}{c^2} - k^2}. \tag{2.22}$$

The surface boundary condition implies that $B = 0$ while the bottom boundary condition leads to:

$$A\gamma \cos \gamma D = 0. \tag{2.23}$$

where D is the depth of the bottom. Thus, either $A = 0$ (the trivial solution) or we must have

$$\gamma D = \left(m + \frac{1}{2}\right)\pi, \qquad m = 1, 2, \ldots,$$

that is, k must assume particular values,

$$k_m = \sqrt{\frac{\omega^2}{c^2} - \left[\left(m + \frac{1}{2}\right)\frac{\pi}{D}\right]^2}, \qquad m = 1, 2, \ldots. \tag{2.24}$$

The corresponding eigenfunctions are given by,

$$Z_m(z) = \sqrt{\frac{2}{D}} \sin \gamma_m z, \tag{2.25}$$

where we have chosen the constant A so that the modes have unit norm as specified in Eq. (2.9).

Equation (2.24), which relates the frequency ω to the wavenumber k_m, is known as the *dispersion relation*. Plots of ω versus k_m are in turn called the dispersion curves. The quantities $v_p(\omega) = \omega/k_m$ and $v_g(\omega) = d\omega/dk$ are respectively the *phase velocity* and the *group velocity* of the mth mode. The group velocity is associated with the radial speed of propagation for a pulse.

The eigenvalues divide into two classes corresponding to *propagating* and *evanescent* modes depending on whether the argument of the square root in Eq. (2.24) is positive or negative. In either case, the square root admits two values k_m and $-k_m$. The positions of these eigenvalues are indicated schematically in Fig. 2.3 by circles. (Their precise positions depend on the frequency, depth and sound speed.)

For the propagating modes we select the branch which gives an outgoing wave. Since we have suppressed a time dependence of the form $e^{-i\omega t}$ we should take the positive value for k_m. These eigenvalues are indicated by the filled circles lying on the positive real axis in Fig. 2.3.

For the evanescent modes we have to choose between roots of the form ia and $-ia$ where a is a positive real number. These modes have the property of either growing or decaying in range. In order to have a bounded solution we take the branch for which k_m lies in the upper half-plane, i.e. $k_m = ia$ with a positive. These eigenvalues are indicated by the filled circles lying on the positive imaginary axis in Fig. 2.3.

The real eigenvalues have an upper bound ω/c. As we reduce the frequency, the eigenvalues on the real axis slide to the left and up the imaginary axis. At a sufficiently low frequency the first mode will make the transition leaving no propagating modes. The frequency at which this occurs is called the *cut-off* frequency for the waveguide.

As a concrete example, consider the isovelocity problem with sound speed $c = 1500\,\mathrm{m/s}$, depth $D = 100\,\mathrm{m}$, and source frequency $f = 100\,\mathrm{Hz}$. Selected modes are plotted in Fig. 2.4. Note that the mth mode has m zeros.

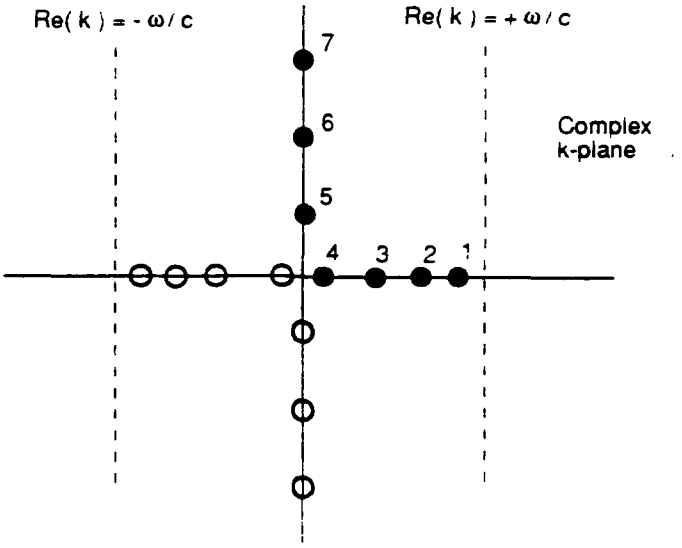


Figure 2.3: Location of eigenvalues for the isovelocity problem.

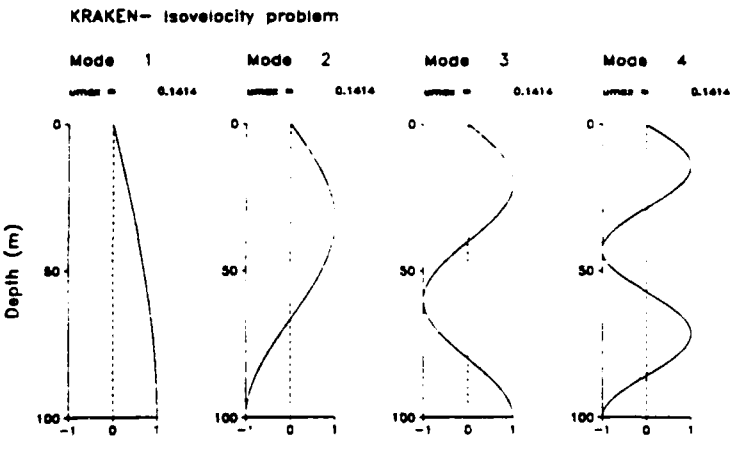


Figure 2.4: Selected modes of the isovelocity problem.

Substituting the formula for the isovelocity modes given in Eq. (2.25) into Eq. (2.15) we obtain a representation of the pressure field:

$$p(r,z) = \frac{i}{2D}\sum_{m=1}^{\infty}\sin(\gamma_m z_s)\sin(\gamma_m z)H_0^{(1)}(k_m r). \tag{2.26}$$

Similarly, from Eq. (2.19) we obtain a representation for the transmission loss as $-10\log I$ where I is an intensity defined by

$$I(r,z) = |p(r,z)|^2 = \left|\frac{1}{D}\sqrt{\frac{8\pi}{r}}\sum_{m=1}^{\infty}\sin(\gamma_m z_s)\sin(\gamma_m z)\frac{e^{ik_m r}}{\sqrt{k_m}}\right|^2. \tag{2.27}$$

In Fig. 2.5 we display the transmission loss for this problem keeping 1, 2 and 3 modes respectively in the modal sum. The source depth is $z_s = 25\,\mathrm{m}$ and the receiver depth is $z = 50\,\mathrm{m}$ in these calculations. Note that as we increase the number of modes the detail in the TL curves also increases. This can be understood by writing the intensity as

$$\begin{aligned} I(r,z) &= \frac{8\pi}{rD^2}\left|\sum_{m=1}^{\infty}A_m e^{ik_m r}\right|^2 \\ &= \frac{8\pi}{rD^2}\sum_{n\neq m}\sum_{m}A_m^2 + A_n^2 + 2A_m A_n\cos(\Delta k_{mn} r) \end{aligned} \tag{2.28}$$

where,

$$\Delta k_{mn} = k_m - k_n. \tag{2.29}$$

and

$$A_m = \frac{\sin(\gamma_m z_s)\sin(\gamma_m z)}{k_m}. \tag{2.30}$$

With just one mode in the series, the complex pressure involves an oscillatory term of the form e^{ikr}, however, its envelope (the intensity) is smooth as indicated in Fig. 2.5. With two modes in the series the intensity is seen to include a term $\cos[(k_1 - k_2)r]$ giving the two-mode interference pattern in Fig. 2.5(b). Note that the interference pattern occurs over a scale significantly larger than the wavelength. Finally, with 3 modes the interference structure shows a further increase in complexity as shown in Fig. 2.5(c).

Many of the properties we see for the isovelocity profile will carry through to more general profiles. On the other hand, while it may still be useful to speak of propagating and evanescent modes, the distinction is blurred when attenuation is included for then all of the modes are displaced into the first quadrant and so all the modes have both a propagating and an evanescent component. Similarly, the cut-off frequency is poorly defined in such cases. These points will be made clearer as we consider more complicated cases.

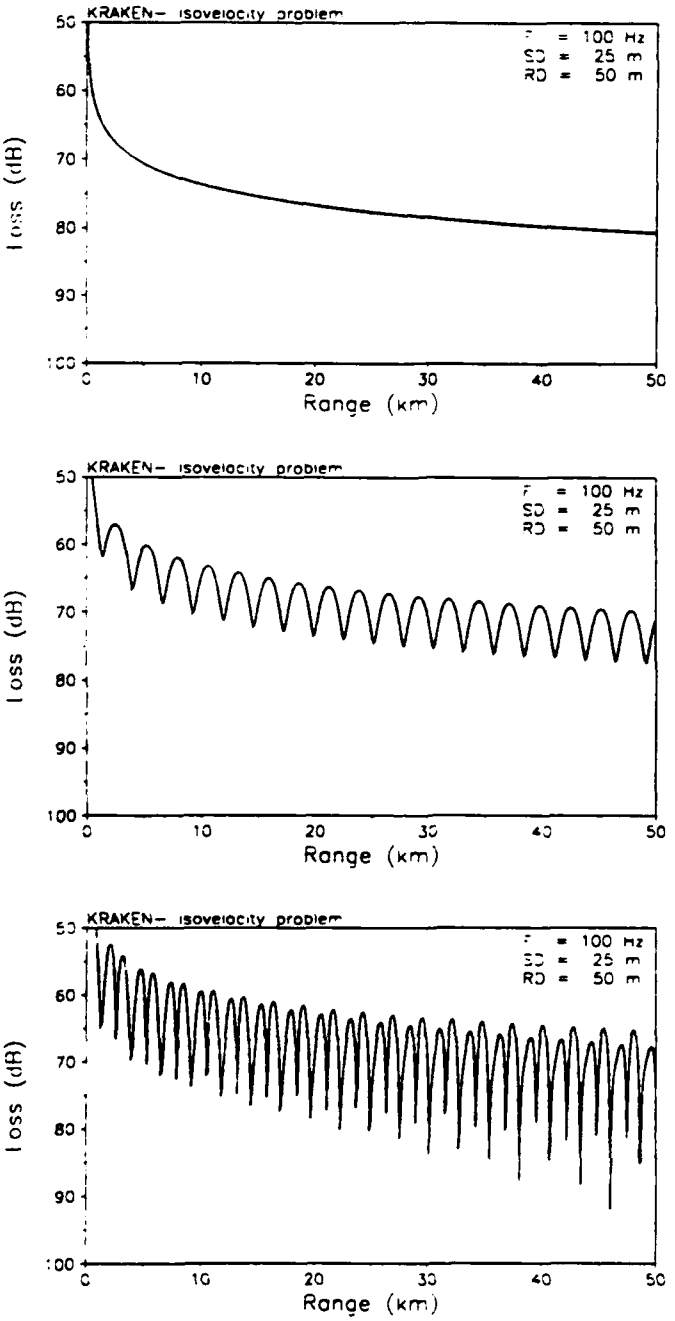


Figure 2.5: Transmission loss for the isovelocity problem using (a) 1 mode, (b) 2 modes and (c) 3 modes.

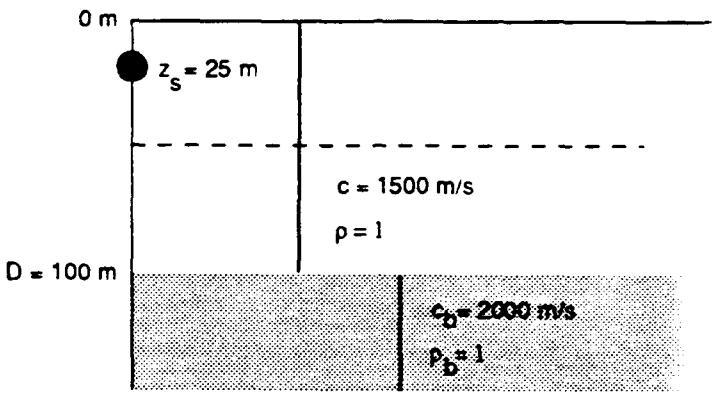


Figure 2.6: Schematic of the Pekeris problem.

2.3 A Generalized Derivation

The derivation of the modal equation which was followed in Sect. 2.2 is inapplicable in many ocean acoustic problems. The key assumption was that after applying the separation of variables we obtained a Sturm-Liouville problem which was non-singular and therefore possessed a complete set of normal modes. As we shall see, even fairly simple scenarios can lead to singular problems for which the normal modes do not form a complete set. More generally, one obtains a mixed spectrum composed of a discrete and a continuous part. The discrete spectrum in such cases leads to a representation involving a sum of modes while the continuous spectrum involves an integral over a continuum of points in k-space.

A simple example of such a problem is provided by the Pekeris waveguide which consists of an isovelocity layer over an isovelocity halfspace. We shall consider the particular problem shown schematically in Fig. 2.6.

Applying the separation of variables technique to this problem we obtain the modal equation (2.7) but with the bottom depth D going to infinity: the modal equation is singular. We can make the domain finite by constructing a boundary condition to be applied at the interface between the two layers. To construct the equivalent boundary condition, we observe that the general solution in the halfspace is given by,

$$Z_{HS}(z) = Be^{-\gamma_b z} - Ce^{\gamma_b z}. \tag{2.31}$$

where,

$$\gamma_b = \sqrt{k^2 - \frac{\omega^2}{c_b^2}}. \tag{2.32}$$

and c_b denotes the sound speed in the bottom. Let us assume for the moment that γ_b is positive. Then, in order to have a bounded solution at infinity, we require C to vanish. At the interface, we require continuity of pressure and normal velocity:

$$Z(D) = Be^{-\gamma_b D}. \tag{2.33}$$

$$\frac{Z'(D)}{\rho} = -B\frac{\gamma_b e^{-\gamma_b D}}{\rho_b}. \tag{2.34}$$

where ρ and ρ_b denote the density in the water and bottom respectively. Dividing these two equations we find that $Z(z)$ must satisfy the boundary condition

$$\frac{\rho Z(D)}{Z'(D)} = -\frac{\rho_b}{\gamma_b(k^2)}. \tag{2.35}$$

Our modal problem then reads:

$$\begin{aligned} \frac{d^2Z(z)}{dz^2} - \left(\frac{\omega^2}{c^2} - k^2\right) Z(z) &= 0, \\ Z(0) &= 0, \\ f(k^2)Z(D) - \frac{g(k^2)}{\rho}Z'(D) &= 0. \end{aligned} \tag{2.36}$$

where,

$$f(k^2) = 1, \qquad g(k^2) = \rho_b \Big/ \sqrt{k^2 - \frac{\omega^2}{c_b^2}}. \tag{2.37}$$

Now we have a modal problem defined on a finite domain, but the boundary condition involves the eigenvalue k^2. Furthermore, the eigenvalue enters through a square root function which introduces a branch cut in the k-plane. Thus, we can convert a modal problem from an infinite domain to a finite domain but the problem remains singular and we are not assured of the completeness of the eigenfunctions.

We shall take another tack which, briefly stated, is to take the spectral integral representation of the solution: close the contour; and calculate the integral as a sum of residues. The terms due to the residues will turn out to correspond to the modes of the problem. Thus, we start with the spectral integral representation[50]

$$\begin{aligned} p(r,z) &= \frac{1}{2\pi}\int_0^\infty G(z,z_s;k)J_0(kr)kdk \\ &= \frac{1}{4\pi}\int_{-\infty}^\infty G(z,z_s;k)H_0^{(1)}(kr)kdk \end{aligned} \tag{2.38}$$

where, $G(z,z_s;k)$ satisfies:

$$\begin{aligned} \rho(z)\left[\frac{1}{\rho(z)}G'(z)\right]' + \left(\frac{\omega^2}{c^2(z)} - k^2\right) G(z) &= \delta(z - z_s), \\ f^T(k^2)G(0) + \frac{g^T(k^2)}{\rho(0)}G'(0) &= 0, \\ f^B(k^2)G(D) + \frac{g^B(k^2)}{\rho(D)}G'(D) &= 0. \end{aligned} \tag{2.39}$$

and primes denote differentiation with respect to z. The top and bottom boundary conditions involve functions $f^{T,B}, g^{T,B}$ representing an angle-dependent impedance. Incidentally, this form allows for fairly complicated bottom types. For instance, Ref. [15] shows how to construct an impedance condition equivalent to an elastic subbottom.

We shall write this problem symbolically as,

$$\mathcal{L}(k_m)G = \delta(z - z_s), \qquad \mathcal{B}_1 G = \mathcal{B}_2 G = 0.$$

The solution of this boundary value problem is given in standard texts (e.g. Ref. [41]) as:

$$G(z, z_s; k) = \frac{-p_1(z_<; k)p_2(z_>; k)}{W(z_s; k)} \tag{2.40}$$

where $z_< = \min(z, z_s)$ and $z_> = \max(z, z_s)$. Furthermore, $W(z; k)$ is the Wronskian:

$$W(z; k) = p_1(z; k)p_2'(z; k) - p_1'(z; k)p_2(z; k).$$

where p_1, p_2 are any non-trivial solutions satisfying the top and bottom boundary conditions respectively. That is,

$$\mathcal{L}(k)p_1 = 0, \qquad \mathcal{B}_1 p_1 = 0, \tag{2.41}$$

$$\mathcal{L}(k)p_2 = 0, \qquad \mathcal{B}_2 p_2 = 0. \tag{2.42}$$

Let us consider a problem such as the Pekeris waveguide which has a single branch cut from a lower halfspace boundary condition. We next close the contour in the spectral integral representation by adding the semicircle C_∞ and the branch cut integral C_{EJP} as shown in Fig. 2.7. (This particular choice of the branch cut which follows the axes is called the EJP cut after Ewing, Jardetsky and Press[42]). Then from Cauchy's integral formula we can write the integral as a sum of residues:

$$\int_{-\infty}^{\infty} - \int_{C_\infty} - \int_{C_{EJP}} = 2\pi i \sum_{m=1}^{M} \text{res}(k_m) \tag{2.43}$$

where $\text{res}(k_m)$ denotes the residue of the mth pole which is enclosed by the contour. These poles are indicated schematically by the filled circles in Fig. 2.7. (Their precise positions depend on the frequency and parameters of the waveguide.) Additional poles, which are not enclosed, will also occur as indicated by the hollow circles. Furthermore, depending on the particular problem and the choice of the branch cut, the number of such residues may be zero, finite or infinite.

As the radius of the semicircle C_∞ goes to infinity, the contribution of that contour goes to zero because the Hankel function decays exponentially as the radius increases. Substituting the representation of the Green's function given in Eq. (2.40) into Eq. (2.38) we then obtain a representation of the field as a sum of residues plus a branch-cut integral, viz.:

$$p(r, z) = \frac{i}{2} \sum_{m=1}^{M} \frac{p_1(z_<; k_m)p_2(z_>; k_m)}{\partial W(z_s; k)/\partial k|_{k=k_m}} H_0^{(1)}(k_m r)k_m - \int_{C_{EJP}} \tag{2.44}$$

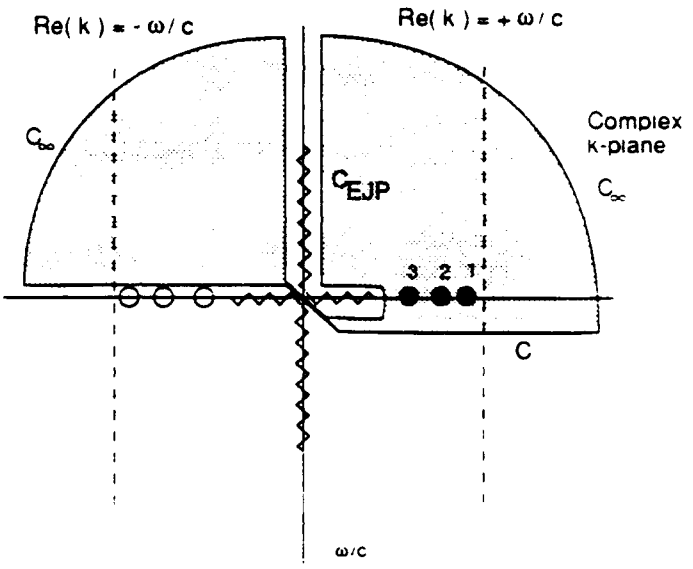


Figure 2.7: Location of eigenvalues for the Pekeris problem (EJP branch cut).

where, k_m is the mth zero of the Wronskian. We arbitrarily order these zeros so that $Re\{k_1\} > Re\{k_2\} > \cdots$. The equation defining the eigenvalues ($W(z_s; k_m) = 0$) is known as the *characteristic equation* or *secular equation*. (In general, any equation whose roots are the eigenvalues will be called a characteristic equation.)

Now if $W(k_m) = 0$ then $p_{1,2}(z; k_m)$ are linearly dependent and we can simply scale them so that they are equal. We shall therefore define $Z_m(z) = p_1(z; k_m) = p_2(z; k_m)$ which satisfies,

$$\mathcal{L}(k_m)Z_m = 0. \qquad \mathcal{B}_1 Z_m = \mathcal{B}_2 Z_m = 0. \tag{2.45}$$

This is, of course, the standard modal equation. If k_m and $Z_m(z)$ form a non-trivial solution of this modal equation, then k_m is a zero of the Wronskian and vice versa. In terms of Z_m we can write

$$p(r, z) = \frac{i}{2} \sum_{m=1}^{M} \frac{Z_m(z_s)Z_m(z)}{\partial W(z_s; k)/\partial k|_{k=k_m}} H_0^{(1)}(k_m r)k_m - \int_{C_{EJP}}. \tag{2.46}$$

This representation of the pressure field is somewhat inconvenient since it requires the evaluation of $\partial W/\partial k$ which is defined in terms of functions $p_{1,2}(z; k)$ that may not be readily available in a particular numerical scheme. In order to simplify this expression, we seek an alternate form for $\partial W/\partial k$. The following result is derived in the next subsection:

$$\frac{\partial W/\partial k}{\rho(z_s)}_{k_m} = 2k_m \int_0^D \frac{Z_m^2(z)}{\rho(z)} dz - \frac{d(f/g)^T}{dk}_{k_m} Z_m^2(0) - \frac{d(f/g)^B}{dk}_{k_m} Z_m^2(D). \tag{2.47}$$

By properly scaling $Z_m(z)$ we can make $\partial W(z_s;k)/\partial k|_{k=k_m} = 1$. Thus we obtain our final representation for the pressure field as

$$p(r,z) = \frac{i}{4\rho(z_s)} \sum_{m=1}^{M} Z_m(z_s) Z_m(z) H_0^{(1)}(k_m r) - \int_{C_{BJP}} . \tag{2.48}$$

where the modes are normalized such that

$$\int_0^D \frac{Z_m^2(z)}{\rho(z)} dz - \frac{1}{2k_m} \frac{d(f/g)^T}{dk}\bigg|_{k_m} Z_m^2(0) + \frac{1}{2k_m} \frac{d(f/g)^B}{dk}\bigg|_{k_m} Z_m^2(D) = 1. \tag{2.49}$$

(An alternate derivation of this result is given by Bucker[43] for the constant density problem.)

It appears we have only made the problem more complicated since we have converted the original spectral integral form to one involving a sum of modes plus another integral term. In practice, however, the branch cut integral can generally be neglected if we are sufficiently far from the source.

The particular nature of the boundary conditions is important in determining the representation. As we have seen, if the upper boundary is a pressure release boundary and the lower boundary is perfectly rigid, then there are no branch-cut contributions: the solution is represented entirely as an infinite sum of modes. In problems with an elastic halfspace, there will be branch-cut terms associated with both S- and P-wave velocities in the halfspace.

Furthermore, the number of terms in the residue series depends on the particular branch cut taken. For instance, if we take the Pekeris branch-cut shown in Fig. 2.8 then it turns out that an additional (typically infinite) set of poles is exposed. These poles are represented by the filled circles numbers 4–6 in the figure. The poles in this second set lie off the real axis in the first quadrant and as a result decay exponentially in range. For this reason the corresponding modes are referred to as *leaky* modes. Thus, we can obtain an infinite variety of representations of the field depending on the choice of branch cut.

In principle, the Pekeris cut offers an advantage in exposing the leaky modes for, as we shall see in the next section, by including the leaky modes we can obtain a solution which is more accurate in the near field. In practice, it is somewhat difficult to reliably locate the leaky modes so the potential gain may come at the expense of robustness in the model. In addition, the leaky modes *grow* exponentially in depth and at some ranges and depths yield a diverging series. Alternatively, it is also possible to calculate the branch cut term numerically as discussed by Stickler [11].

In order to clarify some of these points, let us return to the Pekeris waveguide problem. The solution in the ocean layer which satisfies the pressure release surface condition is given by

$$Z(z) = A \sin \gamma z, \tag{2.50}$$

where,

$$\gamma = \sqrt{\frac{\omega^2}{c^2} - k^2}. \tag{2.51}$$

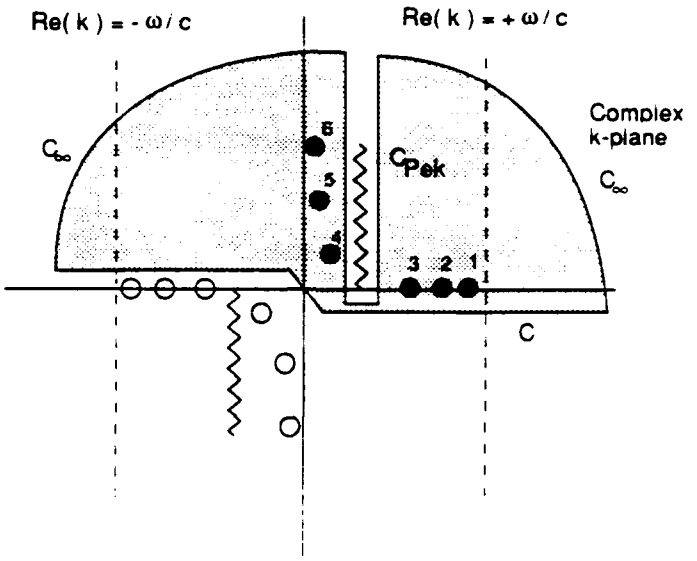


Figure 2.8: Location of eigenvalues for the Pekeris problem (Pekeris branch cut).

In order to obtain a non-trivial solution that satisfies the bottom boundary conditions we must have

$$-\tan\gamma D = \frac{\gamma}{\gamma_b}. \tag{2.52}$$

This is a transcendental equation for the eigenvalues $k_m(\omega)$.

In Fig. 2.9 we have plotted selected modes of the Pekeris problem. We see that modes 1 and 4 are qualitatively similar to the previous isovelocity problem: the solution in the water column is again a sinusoid however the vertical wavenumber is different due to the change in phase associated with the bottom reflection coefficient. Also plotted in Fig. 2.9 are modes 10 and 12 which are leaky modes and therefore manifest a non-zero imaginary part as shown by the dashed line.

Neglecting the cylindrical spreading term, the contribution of an individual mode to the pressure field is proportional to

$$p = \left(e^{i\gamma_m z} - e^{-i\gamma_m z}\right) e^{ik_m r}.$$

Thus, the modes can be thought of as consisting of an up and downgoing plane-wave with an angle of propagation θ defined by $\tan\theta = k_m/\gamma_m$. The branch point occurs at $k = \omega/c_b$ which in the angle domain corresponds precisely to the critical angle. Thus, the modes whose angles are less than the critical angle are trapped, that is, radiate no energy into the halfspace. The leaky modes however have angles above the critical angle and lose energy into the lower halfspace.

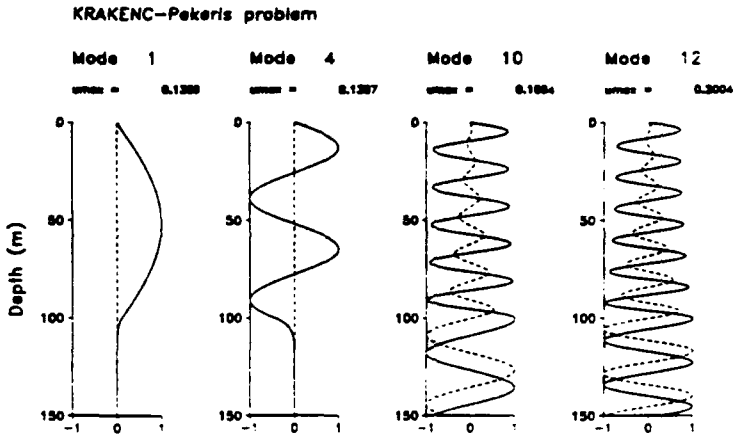


Figure 2.9: Selected modes of the Pekeris problem.

2.3.1 Derivation of the Normalization Formula

In the previous section we used an expression for $\partial W/\partial k$ necessary for normalizing the modes which for completeness we shall now derive. We consider the problem:

$$\begin{aligned} \rho(z)\left[\frac{1}{\rho(z)}G'(z)\right]' + \left(\frac{\omega^2}{c^2(z)} - k^2\right)G(z) &= \delta(z - z_s). \\ f^T(k^2)G(0) + \frac{g^T(k^2)}{\rho(0)}G'(0) &= 0, \\ f^B(k^2)G(D) + \frac{g^B(k^2)}{\rho(D)}G'(D) &= 0. \end{aligned} \tag{2.53}$$

where primes denote differentiation with respect to z. We shall write this problem symbolically as,

$$\mathcal{L}(k_m)g = \delta(z - z_s). \qquad \mathcal{B}_1 g = \mathcal{B}_2 g = 0.$$

The Wronskian is defined by:

$$W(z;k) = p_1(z;k)p_2'(z;k) - p_1'(z;k)p_2(z;k).$$

where p_1, p_2 are any non-trivial solutions that satisfy the top and bottom boundary conditions respectively. That is,

$$\mathcal{L}(k)p_1 = 0. \qquad \mathcal{B}_1 p_1 = 0. \tag{2.54}$$

$$\mathcal{L}(k)p_2 = 0, \qquad \mathcal{B}_2 p_2 = 0. \tag{2.55}$$

Let Z_m be a solution of the unforced boundary value problem.

$$\mathcal{L}(k_m)Z_m = 0, \qquad \mathcal{B}_1 Z_m = \mathcal{B}_2 Z_m = 0.$$

Then,

$$p_2\mathcal{L}(k_m)Z_m - Z_m\mathcal{L}(k)p_2 = 0. \tag{2.56}$$

or, equivalently,

$$\left[p_2\left(\frac{Z_m'}{\rho}\right)' - Z_m\left(\frac{p_2'}{\rho}\right)'\right] + (k^2 - k_m^2)\frac{Z_m p_2}{\rho} = 0. \tag{2.57}$$

This can also be written:

$$\left(p_2\frac{Z_m'}{\rho} - Z_m\frac{p_2'}{\rho}\right)' + (k^2 - k_m^2)\frac{Z_m p_2}{\rho} = 0. \tag{2.58}$$

Taking the integral then gives:

$$\left[\frac{p_2 Z_m' - Z_m p_2'}{\rho}\right]\Bigg|_0^D + (k^2 - k_m^2)\int_0^D \frac{Z_m p_2}{\rho}dz = 0 \tag{2.59}$$

We shall need two intermediate results giving the value of the term in square brackets at $z = 0$ and $z = D$. To obtain the value at $z = 0$ we note that $W(z)/\rho(z)$ is constant since,

$$\left(\frac{W}{\rho}\right)' = \left(\frac{p_1 p_2' - p_1' p_2}{\rho}\right)' = \frac{p_1\mathcal{L}p_2 - p_2\mathcal{L}p_1}{\rho} = 0. \tag{2.60}$$

Thus, we can write:

$$\frac{W(z_s)}{\rho(z_s)} = \frac{W(0)}{\rho(0)} = \frac{p_1(0)p_2'(0) - p_1'(0)p_2(0)}{\rho(0)}. \tag{2.61}$$

and solving for $p_2'(0)$ one obtains,

$$p_2'(0) = \frac{p_1'(0)p_2(0) + W(z_s)\rho(0)/\rho(z_s)}{p_1(0)}. \tag{2.62}$$

This enables us to write,

$$\left.\frac{p_2 Z_m' - Z_m p_2'}{\rho}\right|_0 = \frac{p_2(0)Z_m'(0) - p_1'(0)p_2(0)Z_m(0)/p_1(0)}{\rho(0)} - \frac{W(z_s)}{\rho(z_s)}. \tag{2.63}$$

We can eliminate the derivatives from this equation using the upper boundary condition:

$$\begin{aligned} Z_m'(0) &= -\frac{\rho(0)f^T(k_m)}{g^T(k_m)}Z_m(0) \\ p_1'(0) &= -\frac{\rho(0)f^T(k)}{g^T(k)}p_1(0), \end{aligned} \tag{2.64}$$

Thus Eq. (2.63) becomes:

$$\left.\frac{p_2 Z_m' - Z_m p_2'}{\rho}\right|_0 = -\left[\frac{f^T(k)}{g^T(k)} - \frac{f^T(k_m)}{g^T(k_m)}\right]p_2(0)Z_m(0) - \frac{W(z_s)}{\rho(z_s)}. \tag{2.65}$$

This gives us the value of the term in square brackets in Eq. (2.59) evaluated at $z = 0$. The value at $z = D$ is can be written down directly as:

$$\left.\frac{p_2 Z_m' - Z_m p_2'}{\rho}\right|_D = -\left[\frac{f^B(k)}{g^B(k)} - \frac{f^B(k_m)}{g^B(k_m)}\right] p_2(D) Z_m(D). \tag{2.66}$$

where we have used the bottom boundary condition,

$$\begin{aligned} Z_m'(D) &= -\frac{\rho(D) f^B(k_m)}{g^B(k_m)} Z_m(D) \\ p_1'(D) &= -\frac{\rho(D) f^B(k)}{g^B(k)} p_1(D). \end{aligned} \tag{2.67}$$

Using the results of Eqs. (2.65) and (2.66) in Eq. (2.59) we obtain,

$$\begin{aligned} -\left[\frac{f^B(k)}{g^B(k)} - \frac{f^B(k_m)}{g^B(k_m)}\right] p_2(D) Z_m(D) &- \left[\frac{f^T(k)}{g^T(k)} - \frac{f^T(k_m)}{g^T(k_m)}\right] p_2(0) Z_m(0) \\ -\frac{W(z_s;k) - W(z_s;k_m)}{\rho(z_s)} &+ (k^2 - k_m^2) \int_0^D p_2 \frac{Z_m}{\rho} dz = 0 \end{aligned} \tag{2.68}$$

where we have added in the term $W(z_s; k_m)$. This is permissible since $W(z; k_m) = 0$, that is, the Wronskian vanishes when k is an eigenvalue.

The functions $p_{1,2}(z;k)$ and $Z_m(z)$ may all be scaled freely and still satisfy their respective governing equations. Therefore, without loss of generality, we take $p_2(D;k) = Z_m(D)$. Now, dividing both sides of the equation by $k - k_m$ and taking the limit as $k \to k_m$ we obtain the final result:

$$\left.\frac{\partial W/\partial k}{\rho(z_s)}\right|_{k_m} = 2k_m \int_0^D \frac{[Z_m(z)]^2}{\rho(z)} dz - \left.\frac{d(f/g)^T}{dk}\right|_{k_m} [Z_m(0)]^2 - \left.\frac{d(f/g)^B}{dk}\right|_{k_m} [Z_m(D)]^2. \tag{2.69}$$

2.4 A Deep Water Problem: the Munk Profile

The Munk profile[47] is an idealized sound speed profile: however, it allows us to illustrate many features that are typical of deep-water SSP's. In its general form, the profile is given by

$$c(z) = 1500.0 \left[1.0 - \epsilon\left(\tilde{z} - 1 - e^{-\tilde{z}}\right)\right]. \tag{2.70}$$

The quantity ϵ is taken to be

$$\epsilon = 0.00737$$

while the scaled depth $\tilde{z}$ is given by

$$\tilde{z} = \frac{2(z - 1300)}{1300}.$$

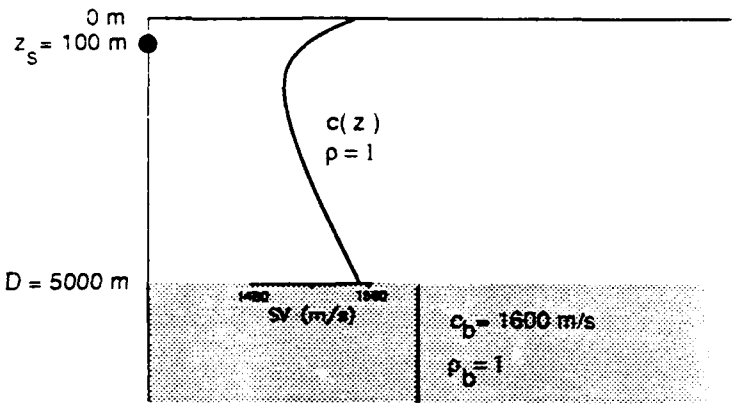


Figure 2.10: Schematic of the deep-water problem.

The resulting profile is plotted in Fig. 2.10. The bottom is taken to lie at a depth of 5000 m. In addition, the bottom sound speed is 1600 m/s and the bottom density is 1 g/cm^3. Taking a source frequency of 50 Hz, we then obtain the modes shown in Fig. 2.11. (An analytic solution for the modes is not available in this case so we use a numerical technique as described in Chap. 3.) Notice that the mode shapes are no longer perfect sinusoids, however the mth mode still has m zero crossings. In addition, the modes are oscillatory near the sound channel axis and exponentially decaying in a layer near the surface and near the bottom. The size of the oscillatory region is larger for the higher order modes.

Some insight into the behavior of the modes can be obtained from the WKB approximation[48]. The WKB approximation to the eigenfunctions is given by

$$Z(z) \approx A \frac{e^{i\int_0^z \gamma(s)\,ds}}{\sqrt{\gamma(z)}} - B \frac{e^{-i\int_0^z \gamma(s)\,ds}}{\sqrt{\gamma(z)}}, \tag{2.71}$$

where

$$\gamma^2(z) = \frac{\omega^2}{c^2(z)} - k_m^2.$$

Thus, locally the solution assumes the oscillating form of sines and cosines near the sound channel axis (where γ is real) and transitions to a solution involving exponentially growing and decaying functions near the surface and bottom (where γ is imaginary). The depths where this transition occurs are the ***turning points*** and are precisely defined by depths where $\gamma^2(z) = 0$. In addition, the amplitude term is seen to be governed by $1/\gamma(z)$ so that as we move away from the sound channel axis (where γ is large) towards the turning point (where γ is small) the amplitude tends

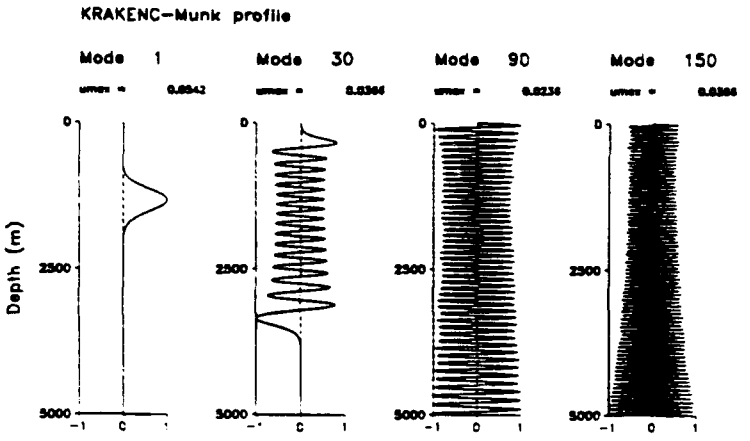


Figure 2.11: Selected modes for the deep-water problem.

to increase. At the turning point the WKB approximation is actually singular. The correct solution, however, has a smooth behavior in the transition layer as may be seen in Fig. 2.11.

In Fig. 2.12 we illustrate the effect of using different numbers of modes to calculate transmission loss. The source depth is chosen to be 100 m. Figure 2.12(a) includes only the *waterborne* modes, that is modes which have their lowest turning point above the ocean bottom. These modes (modes 1 to 63) are exponentially decaying below this turning point and in this sense are not bottom interacting. In ray terms these modes correspond to paths which are refracted away from the bottom.

The transmission loss shows a convergence zone type of pattern involving a beam of energy that emerges from the source and refracts under the influence of the ocean sound speed profile. Since we are using a restricted number of modes, we in effect are producing an angle-limited source. We observe that the transmission loss shows large shadow zones where the acoustic field is negligible. These quiet zones result not from the depth-dependence of the eigenfunctions but from the phasing in range which causes the modes to add up destructively.

In Fig. 2.12(b) we have added in the *bottom bounce* modes (modes 64 to 102). These modes have no turning point and correspond to ray paths which strike the bottom at a subcritical angle. Including these modes effectively widens the source beam-width. A second beam is now visible emanating from the source and reflecting off the bottom.

In Fig. 2.12(c) we have added in a large number of *leaky* modes (modes 103 to 400). As discussed earlier these modes are leaky in the sense that they are displaced from the real axis and therefore lose energy as a function of range. In ray terms, they corresponds to paths which strike the bottom above the critical angle and therefore

are very weakly reflected. In the transmission loss plot we can see that the source angle has now been opened up to 90° revealing the full Lloyd mirror pattern of beams. (The Lloyd mirror results from constructive and destructive interference between the source and its image reflected in the ocean surface.)

2.5 Elastic Media

As discussed in Ref. [15] the elastic quantities (stresses and displacements) satisfy a fourth-order system of ordinary differential equations. We first introduce the stress-displacement vector, $\mathbf{r}$ defined by

$$(r_1, r_2, r_3, r_4) = \left(\frac{u}{ik}, w, \frac{\tau_{xx}}{ik}, \tau_{zz} \right) \tag{2.72}$$

where u is the horizontal displacement, w is the vertical displacement, τ_{xx} is the tangential stress and τ_{zz} is the normal stress. The purpose of introducing the scaling of u and τ_{xx} given in Eq. (2.72) is to eliminate complex quantities from the governing equations and to obtain a form where the eigenvalue k occurs only in squared form. The stress-displacement vector then satisfies

$$\mathbf{r}' = \mathbf{E}\mathbf{r} \tag{2.73}$$

where,

$$\mathbf{E}(z,k) = \begin{bmatrix} 0 & -1 & 1/(\rho c_s^2) & 0 \\ k^2\eta(z) & 0 & 0 & 1/(\rho c_p^2) \\ k^2\zeta(z) - \rho\omega^2 & 0 & 0 & -\eta(z) \\ 0 & -\rho\omega^2 & k^2 & 0 \end{bmatrix} \tag{2.74}$$

where the quantities $\eta(z)$ and $\zeta(z)$ are defined by

$$\eta(z) = \frac{c_p^2 - 2c_s^2}{c_p^2}, \qquad \zeta(z) = \frac{\rho\left[c_p^4 - (c_p^2 - 2c_s^2)^2\right]}{c_p^2} \tag{2.75}$$

and c_p, c_s denote the P and S wave velocities respectively. In this form certain properties of elastic waves are immediately obvious. For instance, since the eigenvalue occurs only as a squared quantity the the eigenvalues will come in pairs. That is, if $(k_j, \mathbf{r}_j)$ is an eigensolution then $(-k_j, \mathbf{r}_j)$ is also an eigensolution.

The above equations for $\mathbf{r}$ are combined with interfacial and boundary conditions to completely specify the acousto-elastic modal problem. At an elastic-elastic interface, one requires continuity of $\mathbf{r}$ (i.e., continuity of displacements and stresses). At an acousto-elastic interface the condition of continuity of horizontal displacement is relaxed. Noting that 1) pressure is the negative of the normal stress, τ_{zz}, 2) τ_{xx} vanishes in an acoustic medium, and 3) the gradient of the pressure gives the time derivative of the velocity field, one obtains,

$$\begin{aligned} \omega^2 r_2(z) &= p'(z) \\ r_3(z) &= 0 \\ r_4(z) &= -p(z). \end{aligned} \tag{2.76}$$

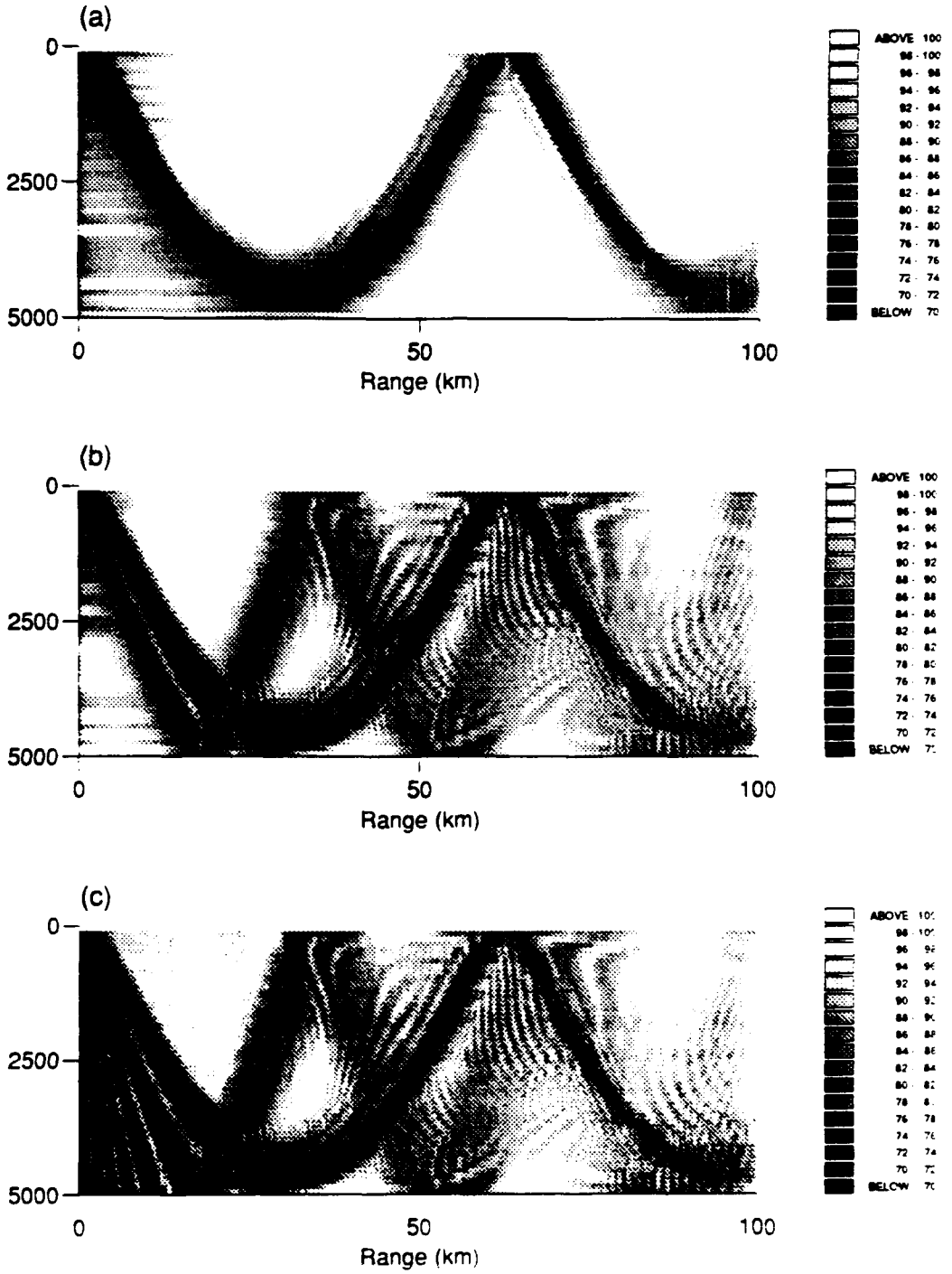


Figure 2.12: Transmission loss for the deep-water problem inciuding (a) waterborne modes only. (b) bottom bounce modes and (c) leaky modes.

KRAKEN and KRAKENC use the reduced delta-matrix formulation. This is obtained by introducing a new set of dependent variables defined by,

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} r_1 s_2 - r_2 s_1 \\ r_3 s_4 - r_4 s_3 \\ r_1 s_3 - r_3 s_1 \\ r_2 s_3 - r_3 s_2 \\ r_1 s_4 - r_4 s_1 \\ r_2 s_4 - r_4 s_2 \end{bmatrix}, \tag{2.77}$$

where $\mathbf{r}$ and $\mathbf{s}$ denote two linearly independent solutions in the elastic medium. Note that $\mathbf{y}$ involves all permutations of r_i, s_j with an ordering chosen to obtain a simple form for the $\mathbf{y}$ equations. By differentiating the above equations and substituting into Eq. (2.74) we find that $\mathbf{y}$ satisfies a system of differential equations:

$$\mathbf{y}' = \mathbf{W}\mathbf{y} \tag{2.78}$$

where,

$$\mathbf{W} = \begin{bmatrix} 0 & 0 & 0 & 1/(\rho c_s^2) & -1/(\rho c_p^2) \\ 0 & 0 & 0 & -\omega^2\rho & -(k^2 t(z) - \omega^2\rho) \\ 0 & 0 & 0 & 1 & \eta(z) \\ k^2\zeta(z) - \omega^2\rho & 1/(\rho c_p^2) & -k^2\eta(z) & 0 & 0 \\ \omega^2\rho & -1/(\rho c_s^2) & -2k^2 & 0 & 0 \end{bmatrix} \tag{2.79}$$

The differential equation for y_6 reduces to $y_6 = -k^2 y_5$ and has been eliminated from the system. In terms of the y-functions the interface conditions between the acoustic medium and a stratified elastic bottom can be written as,

$$f^B(k^2) Z_j(D) + g^B(k^2) \frac{dZ_j(D)}{dz} = 0 \tag{2.80}$$

with,

$$\begin{aligned} f^B(k^2) &= \omega^2 y_4, \\ g^B(k^2) &= y_2 \end{aligned} \tag{2.81}$$

2.6 Boundary and Interface Conditions

In the simplest ocean models the ocean surface is modeled as a pressure release surface and the ocean bottom is assumed perfectly rigid. This leads to Dirichlet and Neumann boundary conditions[2] respectively and the modal problem is a conventional Sturm-Liouville eigenvalue problem. Considering the bottom boundary we note that there really is no well-defined bottom depth— below sediment lies basalt and one may continue from mantle to core, ... The truncation of the interval is

[2] A Robin condition on a function $p(z)$ has the form $fp(z) + gp'(z) = 0$. A Dirichlet condition has the form $p(z) = 0$ and a Neumann condition has the form $p'(z) = 0$.

justified when including additional depth no longer results in a significant change in the result. This somewhat nebulous transition occurs when the ocean subbottom is thick enough that material absorption eliminates significant energy return from deeper depths by refraction or reflection.

Mitigating against a conservative policy in carrying depth varying properties to great depths is the increased cost of solving the modal equation on a large domain. Thus it is desirable to truncate the problem at the shallowest possible depth. The rigid bottom model makes sense at a sediment/basalt interface where there is a strong impedance contrast. Basalts, however, are typically characterized by a strong elastic wave speed gradient which refracts ray paths back into the ocean. At mid- to high-frequencies, say above 50 Hz, this refracted energy will be severely attenuated and may be safely ignored. A more realistic bottom boundary condition is obtained with an acoustic half-space.

The boundary conditions corresponding to these various cases are provided below. The results are presented in three forms: 1) as a Robin condition on the pressure, 2) as boundary conditions on the stress-displacement vector $\mathbf{r}$ and 3) as boundary conditions on the solvability vector $\mathbf{y}$. The first form is used for problems where all internal media are acoustic; the second form is used for problems where some internal media are elastic; the third form is used when elastic displacements are required (KRAKEL).

2.6.1 Perfectly Free Boundary (Dirichlet BC)

The free surface condition is an approximation for the surface boundary condition. (A more sophisticated boundary condition is obtained by using a homogeneous acoustic half-space to model the atmosphere above the ocean. The impedance contrast is, however, so large that there is no practical need for modeling the atmosphere.) For an acoustic medium this yields,

$$p(0) = 0 \tag{2.82}$$

while for an elastic medium both stresses must vanish

$$r_3(0) = r_4(0) = 0. \tag{2.83}$$

Thus, two linearly independent solutions can be obtained using.

$$\begin{aligned} \mathbf{r} &= (1,0,0,0) \\ \mathbf{s} &= (0,1,0,0) \end{aligned} \tag{2.84}$$

From the definitions of $\mathbf{y}$ in Eq. (2.77) we obtain the following boundary conditions.

$$\mathbf{y} = (0,1,0,0,0). \tag{2.85}$$

2.6.2 Perfectly Rigid Boundary (Neumann BC)

The perfectly rigid assumption enjoys some popularity for a bottom boundary. For a purely acoustic problem, this becomes.

$$p'(D) = 0 \tag{2.86}$$

For an elastic medium, this implies that both displacements must vanish,

$$r_1(D) = r_2(D) = 0. \tag{2.87}$$

Thus, two linearly independent solutions can be obtained using,

$$\begin{aligned} \mathbf{r} &= (0,0,1,0) \\ \mathbf{s} &= (0,0,0,1). \end{aligned} \tag{2.88}$$

From the definitions of $\mathbf{y}$ in Eq. (2.77) we obtain the following boundary conditions,

$$\mathbf{y} = (1,0,0,0,0). \tag{2.89}$$

2.6.3 Acoustic Half-space Conditions (Robin BC)

We consider first an acoustic bottom half-space characterized by a single wave speed, c_{pb} and a density, ρ_b. The general solution in the half-space is given by,

$$Z_{HS}(z) = Ae^{-\gamma_b z} + Be^{\gamma_b z}. \tag{2.90}$$

where,

$$\gamma_b = \sqrt{k^2 - \frac{\omega^2}{c_{pb}^2}}, \tag{2.91}$$

and the Pekeris branch of the square root is used to expose the leaky modes. In order to have a bounded solution at infinity, we require B to vanish. At the interface, we require continuity of pressure and normal displacement which implies,

$$Z(D) = Ae^{-\gamma_b D} \tag{2.92}$$

$$Z'(D) = -A\gamma_b e^{-\gamma_b D} \tag{2.93}$$

Thus, we obtain the bottom impedance condition.

$$Z(D) - \frac{\rho_b}{\gamma_b} Z'(D) = 0. \tag{2.94}$$

A similar procedure yields the result for a top homogeneous half-space.

$$Z(D) - \frac{\rho_t}{\gamma_t} Z'(D) = 0. \tag{2.95}$$

which differs by a sign change. Note that by letting $\rho \to 0$ we obtain the free-surface boundary condition and $\rho \to \infty$ gives the perfectly rigid boundary condition.

2.6.4 Elastic Half-space Conditions

The solution for a homogeneous elastic medium is given in terms of P- and S-wave potentials, ϕ and ψ respectively. For a bounded solution these potentials take the form:

$$\phi(z) = Ae^{-\gamma_p z}, \qquad \psi(z) = Be^{-\gamma_s z} \tag{2.96}$$

where,

$$\gamma_p = \sqrt{k^2 - \frac{\omega^2}{c_{pb}^2}}, \qquad \gamma_s = \sqrt{k^2 - \frac{\omega^2}{c_{sb}^2}}. \tag{2.97}$$

In terms of these potentials, the elastic displacements are given by,

$$u = \phi_x - \psi_z, \qquad w = \phi_z + \psi_x. \tag{2.98}$$

So in terms of u and w we can write the most general form of the half-space solution as:

$$\begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} ik & -\gamma_s ik \\ -\gamma_p & -k^2 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \tag{2.99}$$

Recall.

$$(r_1, r_2) = \left(\frac{u}{ik}, w\right) \tag{2.100}$$

and from Eq. (2.74) we obtain,

$$r_3 = \mu(r_1' + r_2) \tag{2.101}$$

$$r_4 = (\lambda - 2\mu)r_2' - k^2\lambda r_1 \tag{2.102}$$

so that the most general solution in the lower half-space is

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} 1 & \gamma_s \\ -\gamma_p & -k^2 \\ -2\mu\gamma_p & -\mu(\gamma_s^2 + k^2) \\ \mu(\gamma_s^2 + k^2) & 2\mu\gamma_s k^2 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \tag{2.103}$$

Taking the columns of the above matrix as two linearly independent solutions and substituting into the the definitions of y in Eq. (2.77) we obtain the following boundary conditions.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} (\gamma_s\gamma_p - k^2)\ \mu \\ ((\gamma_s^2 - k^2)^2 - 4\gamma_s\gamma_p k^2)\,\mu \\ 2\gamma_s\gamma_p - \gamma_s^2 - k^2 \\ \gamma_p(k^2 - \gamma_s^2) \\ \gamma_s(\gamma_s^2 - k^2) \end{bmatrix} \tag{2.104}$$

Note that the classical dispersion relation for Rayleigh waves is obtained by taking the free surface condition $y_2 = 0$.

2.6.5 Tabulated Reflection Coefficients

In some cases, it may be preferred to characterize the ocean surface or bottom by a complex reflection coefficient as a function of grazing angle. Such a reflection coefficient may be specified by tabulating the magnitude and phase at a discrete set of θ-values. To completely define the normal mode problem, $R(\theta)$ must be defined continuously for some range of θ-values. A routine (REFCO) accomplishes this by linearly interpolating the magnitude and phase of the reflection coefficient. In addition, the reflection coefficient must be converted to an equivalent Robin style boundary condition with coefficients which are a function of horizontal propagation number, k.

Note that a surface reflection coefficient asserts a relationship between an upgoing and downgoing wave of the following form.

$$p(z) = e^{i\gamma z} + Re^{-i\gamma z} \tag{2.105}$$

where

$$\gamma = \sqrt{\frac{\omega^2}{c^2(0)} - k^2} \tag{2.106}$$

is the vertical wavenumber. Then,

$$p'(z) = i\gamma \left(e^{i\gamma z} + Re^{-i\gamma z} \right) \tag{2.107}$$

which implies that,

$$\frac{p}{p_z}\Big|_{z=0} = \frac{1}{i\gamma}\frac{1+R}{1-R}. \tag{2.108}$$

Thus, the reflection coefficient implies an impedance or Robin style boundary condition of the form.

$$p - \frac{1}{i\gamma}\frac{1+R}{1-R}p' = 0. \tag{2.109}$$

For bottom reflection we use the form

$$p(z) = e^{-i\gamma z} - Re^{i\gamma z} \tag{2.110}$$

which leads to,

$$p + \frac{1}{i\gamma}\frac{1+R}{1-R}p' = 0. \tag{2.111}$$

In general, the reflection coefficient is a function of the grazing angle θ which is related to γ by

$$\tan\theta = \frac{\gamma}{k}. \tag{2.112}$$

2.7 Loss Mechanisms

2.7.1 Material Absorption

Equations for material absorption in the water column are summarized by Urick[53]. A somewhat standard form due to Thorpe is:

$$\alpha^{Thorpe} = \frac{40f^2}{4100+f^2} + \frac{0.1f^2}{1+f^2} \tag{2.113}$$

where f is the frequency in kilohertz and α^{Thorpe} is in units of dB/m. Attenuation in sediments is somewhat more complicated and must be specified directly.

Material absorption is included by adding an imaginary part to the sound speed so that $c(z) = c_r(z) + ic_i(z)$. Typically, the attenuation is specified in more physical units so it is necessary to perform a conversion. In the following subsections we present the results for various cases.

α in nepers/m

If α is specified in nepers/m then we expect a plane wave to decay in the form

$$e^{-\alpha r}. \tag{2.114}$$

In free space, the solution takes the form,

$$e^{-i\omega/c} = e^{-i\omega\frac{c_r - ic_i}{c_r^2+c_i^2}}. \tag{2.115}$$

Thus,

$$c_i = \frac{\alpha |c|^2}{\omega}. \tag{2.116}$$

All other attenuation units will be represented in terms of α in nepers/m.

$\alpha^{(m)}$ in dB/meter

We require that the ratio of the intensity in dB between points one meter apart be given by $\alpha^{(m)}$. That is,

$$\alpha^{(m)} = -10\log_{10}\frac{I(r-1)}{I(r)} = -20\log_{10}\frac{e^{-\alpha(r-1)}}{e^{-\alpha r}} = 20\alpha\log_{10}e \tag{2.117}$$

which implies

$$\alpha \approx \frac{\alpha^{(m)}}{8.69}. \tag{2.118}$$

$\alpha^{(w)}$ in dB/wavelength

The attenuation in dB/m is given by $\alpha^{(m)} = \alpha^{(w)}/\lambda$. Therefore,

$$\alpha \approx \frac{\alpha^{(w)}}{8.69\lambda}. \tag{2.119}$$

Table 2.1: Approximate conversion factors between attenuation units.

	desired units				
user units	α	$\alpha^{(m)}$	$\alpha^{(w)}$	$\alpha^{(f)}$	Q^{-1}
α	1	8.69	8.69λ	$8686/f$	$\lambda/3.14$
$\alpha^{(m)}$	0.115	1	$\lambda/1000$	$1000/f$	0.037λ
$\alpha^{(w)}$	$0.115/\lambda$	$1000/\lambda$	1	$1000/c$	0.037
$\alpha^{(f)}$	$f/8686$	$f/1000$	$c/1000$	1	$c/27288$
Q^{-1}	$3.14/\lambda$	$0.115/\lambda$	27.29	$27288/c$	1

$\alpha^{(f)}$ in dB/(km Hz)

The attenuation in dB/m is given by $\alpha^{(m)} = (f/1000)\alpha^{(f)}$. Thus.

$$\alpha \approx \frac{\alpha^{(f)}}{8.69}\frac{f}{1000}. \tag{2.120}$$

Q

The term 'Q' is defined in different and inconsistent ways. (See the discussion in Aki and Richards [50].) We take,

$$\alpha \equiv \frac{\pi}{\lambda Q}. \tag{2.121}$$

These results are summarized in Table 1.

2.7.2 Twersky Scatter Theory

The Twersky scatter model is used primarily for modeling under-ice scatter effects as suggested by Diachok [51]. In effect the ice is modeled as a free surface with a uniform distribution of cylindrical bosses with elliptical cross-sections which crudely represent ice keels. A reflection coefficient for this rough surface is constructed by combining the effects of the scattering function for each individual cylindrical boss. The scattering function is in turn computed analytically by a modal sum involving Mathieu functions. The program for computing this specular reflection coefficient was developed by Wales[52] for a spectral integral code.

The cylindrical bosses are described in terms of the principal radii of the elliptical cross-section, the linear density, i.e. the number of bosses per kilometer, and whether the bosses are perfectly free or rigid objects. This leads to a reflection coefficient, $R(\theta)$, which is a complex number incorporating both phase and amplitude information. The reflection coefficient is formally valid in the far-field but is applied in an ad hoc fashion at the ice/water interface. This analytical formula for the surface reflection coefficient is then converted to an equivalent Robin boundary condition using Eq. (2.109).

2.7.3 Kirchhoff Scatter Theory

For open ocean surface roughness a simple scatter model based on Kirchhoff theory is implemented. The reflection coefficient is simply,

$$R(\theta) = 1 - 2\gamma^2\sigma^2, \tag{2.122}$$

where σ is the RMS roughness and γ is the vertical wavenumber given by,

$$\gamma^2 = \frac{\omega^2}{c^2} - k^2. \tag{2.123}$$

When $\gamma\sigma = 1/\sqrt{2}$ this formula predicts a vanishing reflection coefficient. It should not be used for larger values of $\gamma\sigma$.

2.7.4 Interfacial Roughness

Kuperman and Ingenito [49] obtained the following interface condition for an interface with RMS roughness σ:

$$\begin{bmatrix} p_2 \\ p_2'/\rho \end{bmatrix} = \sigma^2 A \begin{bmatrix} p_1 \\ p_1'/\rho \end{bmatrix} \tag{2.124}$$

where the elements of A are given by:

$$\begin{aligned} a_{11} &= 0.5(\gamma_1^2 - \gamma_2^2) - \frac{\rho_2\gamma_1^2 - \rho_1\gamma_2^2)(\gamma_1 + \gamma_2)}{\Delta} \\ a_{12} &= \frac{i(\rho_2 - \rho_1)^2\gamma_1\gamma_2}{\Delta} \\ a_{21} &= \frac{-i(\rho_2\gamma_1^2 - \rho_1\gamma_2^2)^2}{\rho_1\rho_2\Delta} \\ a_{22} &= 0.5(\gamma_1^2 - \gamma_2^2) - \frac{(\rho_2 - \rho_1)\gamma_1\gamma_2(\gamma_1 + \gamma_2)}{\Delta} \end{aligned} \tag{2.125}$$

with,

$$\Delta = \rho_1\gamma_2 - \rho_2\gamma_1 \tag{2.126}$$

and,

$$\gamma^2 = \frac{\omega^2}{c^2} - k^2. \tag{2.127}$$

The subscripts 1 and 2 denote properties in the upper and lower media, respectively. For a free surface the appropriate result is obtained by letting ρ and γ go to zero. Note that this formula also breaks down in a catastrophic fashion if $\gamma\sigma$ is too large.

2.7.5 Perturbational Treatment of Loss Mechanisms

Loss may be caused by scatter at boundaries or interfaces or by material absorption. In the former case the loss is manifest as a complex sound speed while in the latter case the interfacial condition is modified. Both of these mechanisms can be handled by straightforward modifications of the numerical algorithm; however, the eigenvalues become complex, requiring the use of complex arithmetic. More importantly, the root-finder must be modified to perform not just a line search on the real axis but a 2-D search in the complex k-plane. While robust and efficient root-finders can be constructed for the real problem, the complex root-finders are failure-prone.

An attractive alternative to complex eigenvalue searches is to compute the real eigenvalues and then obtain an approximation to the imaginary parts using perturbation theory. To illustrate the technique we consider the modal problem with simple pressure-release and rigid-bottom boundary conditions. That is,

$$\begin{aligned} \rho(z)\left[\frac{1}{\rho(z)}Z_m'(z)\right]' + \left(K^2(z) - k_m^2\right) Z_m(z) &= 0, \\ Z_m(0) &= 0, \\ Z_m'(D) &= 0, \\ \int_0^D \frac{Z_m^2(z)}{\rho(z)}dz &= 1, \end{aligned} \tag{2.128}$$

where, $K^2(z) = \omega^2/c^2(z)$. We next write

$$K^2(z) = K_0^2(z) - \epsilon K_1^2(z) - \cdots \tag{2.129}$$

where $K_0^2(z)$ corresponds to the unperturbed sound speed profile which for lossy problems is simply the real part of $K^2(z)$. In the following the subscript indicating mode number will be suppressed. The next step is to seek a solution of the form:

$$Z(z) = Z_0(z) - \epsilon Z_1(z) - \cdots \tag{2.130}$$

and

$$k^2 = k_0^2(z) - \epsilon k_1^2 - \cdots \tag{2.131}$$

Substituting into Eq. (2.128) and collecting terms of like order we obtain:

$$\begin{aligned} O(1): \quad \rho(z)\left[\frac{1}{\rho(z)}Z_0'(z)\right]' - \left(K_0^2(z) - k_0^2\right) Z_0(z) &= 0, \\ Z_0(0) &= 0, \\ Z_0'(D) &= 0, \\ \int_0^D \frac{Z_0^2(z)}{\rho(z)}dz &= 1. \end{aligned} \tag{2.132}$$

This is the lossless eigenvalue problem and can be solved on the real axis. The next higher-order equation is,

$$O(\epsilon): \quad \rho(z)\left[\frac{1}{\rho(z)}Z_1'(z)\right]' - \left(K_0^2(z) - k_0^2\right) Z_1(z) = \left(K_1^2(z) - k_1^2\right) Z_0(z),$$

$$\begin{aligned} Z_1(0) &= 0, \\ Z_1'(D) &= 0, \\ \int_0^D \frac{Z_1^2(z)}{\rho(z)} dz &= 1. \end{aligned} \tag{2.133}$$

From the Fredholm Alternative Theorem, the inhomogeneous term on the right-hand side must be orthogonal to all solutions of the homogeneous adjoint problem in order for a solution to exist[41]. (In terms of a vibrating string, this means that a steady-state solution does not exist when the string is forced at a resonant frequency.) The solutions of the adjoint problem are simply the modes $Z_0(z)$. Therefore, we must have

$$\int_0^D \left(K_1^2(z) - k_1^2\right) \frac{Z_0^2(z)}{\rho(z)} dz = 0, \tag{2.134}$$

which implies,

$$k_1^2 = \frac{-\int_0^D K_1^2(z) Z_0^2(z)/\rho(z) dz}{\int_0^D Z_0^2(z)/\rho(z) dz}. \tag{2.135}$$

This is the first correction due to an arbitrary perturbation $K_1^2(z)$. To apply this formula to material absorption we take the complex sound speed $c(z) = c_r(z) + ic_i(z)$ and form $K^2(z) = K_r^2(z) + iK_i^2(z) = \omega/c(z)$. The real part, K_r^2, is used to generate our zeroth-order unperturbed problem which is easily solved to provide eigenvalues k_r and eigenfunctions $Z(z)$. We next denote the perturbation term by $\epsilon K_1^2 = iK_i^2(z)$ and the corresponding perturbation to the eigenvalue by $\epsilon k_1^2 = ik_i^2$. In this notation, Eq. (2.135) reads

$$\begin{aligned} ik_i^2 &= \epsilon k_1^2 \\ &= \frac{-\int_0^D \epsilon K_1^2(z) Z^2(z)/\rho(z) dz}{\int_0^D Z^2(z)/\rho(z) dz} \\ &= \frac{-\int_0^D iK_i^2(z) Z^2(z)/\rho(z) dz}{\int_0^D Z^2(z)/\rho(z) dz} \end{aligned} \tag{2.136}$$

or,

$$k_i^2 = \frac{-\int_0^D K_i^2(z) Z^2(z)/\rho(z) dz}{\int_0^D Z^2(z)/\rho(z) dz}. \tag{2.137}$$

In practice, this perturbational approximation is usually adequate. It gives poor numerical accuracy when the imaginary part is very large, that is, when the mode decays very rapidly in range. In such cases the accuracy is normally not critical since the pressure field is then usually dominated by other modes which are less severely attenuated.

A similar perturbational approach is also useful for treating loss due to surface or bottom roughness. Kuperman and Ingenito[49] derive the result:

$$k_i^{(scat)} = \frac{\rho\sigma^2\gamma_m}{2k_m} \left(\frac{dZ_m(z)}{dz}\right)^2_{z=0}. \tag{2.138}$$

for the perturbation in the eigenvalues due to surface scatter loss. Here, σ denotes the RMS roughness of the sea surface, and

$$\gamma_m = \sqrt{\frac{\omega^2}{c^2(0)} - k_m^2}. \tag{2.139}$$

2.8 Normal Modes for Range-Dependent Environments

We tend to think of normal mode models as primarily suitable for range-independent problems, however, it is in principle easy to extend them to provide range-dependent solutions. One way of doing this is to divide the range axis into a number of segments and approximate the field as range-independent within each segment. The solution within a range independent segment is constructed using the standard normal mode solution and interface conditions (continuity of pressure and normal velocity) are used to 'glue' the solutions together.

This "coupled mode" approach is straightforward but leads to a computationally intensive procedure. For this reason it is, at least at the present, primarily useful for providing an exact solution for verifying simpler approximate models such as the PE technique. The full two-way solution is not actually implemented in KRAKEN, however, it provides a useful starting point for discussing two successive simplifications in which we ignore 1) the back-scattered component, and 2) coupling between different modes at the interfaces (*the adiabatic approximation*). These approximations are frequently a reasonable compromise between accuracy and run-time.

2.8.1 Coupled Modes

Our derivation follows Evans [65]. We begin by dividing the problem into N segments in range as illustrated in Fig. 2.13. Neglecting contributions from higher order modes or the continuous spectrum, the general solution in the jth segment can be written as follows:

$$p^j(r,z) = \sum_{m=1}^{M} \left[a_m^j \hat{H}1_m^j(r) + b_m^j \hat{H}2_m^j(r) \right] Z_m^j(z), \tag{2.140}$$

where $\hat{H}1,2$ are the following ratios of Hankel functions.

$$\hat{H}1_m^j(r) = \frac{H_0^{(1)}(k_m^j r)}{H_0^{(1)}(k_m^j r_{j-1})}, \tag{2.141}$$

$$\hat{H}2_m^j(r) = \frac{H_0^{(2)}(k_m^j r)}{H_0^{(2)}(k_m^j r_{j-1})}. \tag{2.142}$$

and we define $r_{j-1} = r_1$ in the special case where $j = 1$. This scaling of the Hankel functions is done to avoid overflow problems for the leaky modes. For such modes the Hankel functions involve growing and decaying exponentials. In practice, it is convenient to replace the Hankel functions by their large argument asymptotic

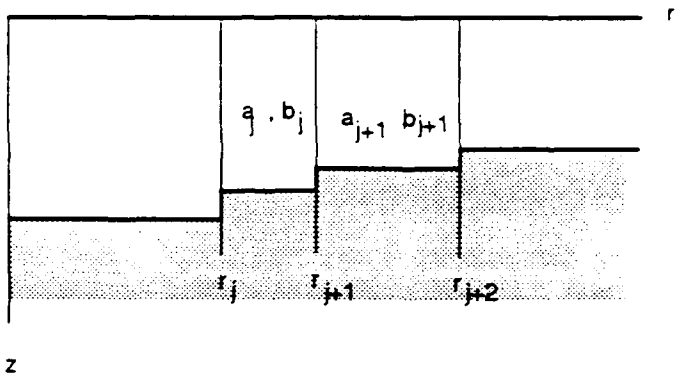


Figure 2.13: Segmentation for coupled mode formulation.

representation yielding:

$$\hat{H}1_m^j(r) \sim H1_m^j(r) = \sqrt{\frac{r_{j-1}}{r}} e^{ik_m^j(r-r_{j-1})}, \tag{2.143}$$

$$\hat{H}2_m^j(r) \sim H2_m^j(r) = \sqrt{\frac{r_{j-1}}{r}} e^{-ik_m^j(r-r_{j-1})}. \tag{2.144}$$

We shall use this asymptotic representation in the remainder of the analysis.

Next we impose continuity of pressure at the jth interface:

$$\sum_{m=1}^{M} \left(a_m^{j+1} + b_m^{j+1}\right) Z_m^{j+1}(z) = \sum_{m=1}^{M} \left[a_m^j H1_m^j(r_j) + b_m^j H2_m^j(r_j)\right] Z_m^j(z). \tag{2.145}$$

This matching condition involves a continuum of depth points in that we require continuity of pressure for all z-values. In practice, however, we are going to work with a limited mode set and therefore we need a finite set of conditions which relate the M mode coefficients a_m, b_m. This can be done in several ways. For instance, we could require continuity of pressure at M discrete depth points. We shall impose a moment condition that the error considered as a function of depth should have vanishing components of each of the first M modes. Thus, we apply the operator

$$\int (\cdot) \frac{Z_l^{j+1}(z)}{\rho_{j+1}(z)} dz, \tag{2.146}$$

to our matching equation where $l = 1, \ldots, M$. Because of the orthogonality property:

$$\int \frac{Z_m^{j+1}(z) Z_l^{j+1}(z)}{\rho_{j+1}(z)} dz = \delta_{lm}. \tag{2.147}$$

only one term remains from the sum on the left of Eq. (2.145). Therefore we have:

$$a_l^{j+1} + b_l^{j+1} = \sum_{m=1}^{M} \left[a_m^j H1_m^j(r_j) + b_m^j H2_m^j(r_j) \right] \tilde{c}_{lm}, \qquad l = 1, \ldots, M, \tag{2.148}$$

where,

$$\tilde{c}_{lm} = \int \frac{Z_l^{j+1}(z) Z_m^j(z)}{\rho_{j+1}(z)} \, dz. \tag{2.149}$$

In matrix notation, we can write this equation as:

$$\mathbf{a}^{j+1} + \mathbf{b}^{j+1} = \tilde{\mathbf{C}}^j \left(\mathbf{H}_1^j \mathbf{a}^j + \mathbf{H}_2^j \mathbf{b}^j \right), \tag{2.150}$$

where, $\mathbf{H}_1^j$ and $\mathbf{H}_2^j$ denote the diagonal matrices with entries $H1_m^j(r_j)$ and $H2_m^j(r_j)$ respectively. In addition, $\tilde{\mathbf{C}}$ is the matrix with entries $\tilde{c}_{lm}$ and $\mathbf{a}, \mathbf{b}$ are column vectors with entries a_l, b_l respectively.

We next impose continui y of radial particle velocity. The particle velocity is proportional to

$$\frac{1}{\rho_j} \frac{\partial p^j(r,z)}{\partial r} \approx \frac{1}{\rho_j} \sum_{m=1}^{M} k_m^j \left[a_m^j H1_m^j(r) - b_m^j H2_m^j(r) \right] Z_m^j(z). \tag{2.151}$$

This time we apply the operator,

$$\int (\cdot) Z_l^{j+1}(z) \, dz, \tag{2.152}$$

to obtain,

$$a_l^{j-1} - b_l^{j+1} = \sum_{m=1}^{M} \left[a_m^j H1_m^j(r_j) - b_m^j H2_m^j(r_j) \right] \hat{c}_{lm}, \qquad l = 1, \ldots, M, \tag{2.153}$$

where,

$$\hat{c}_{lm} = \frac{k_m^j}{k_l^{j+1}} \int \frac{Z_l^{j-1}(z) Z_m^j(z)}{\rho_j(z)} \, dz. \tag{2.154}$$

Note that $\hat{c}$ differs from $\tilde{c}$ in the density term of the integral and by a ratio of horizontal wavenumbers.

In matrix notation, this matching condition can be written as

$$\mathbf{a}^{j+1} - \mathbf{b}^{j+1} = \hat{\mathbf{C}}^j \left(\mathbf{H}_1^j \mathbf{a}^j - \mathbf{H}_2^j \mathbf{b}^j \right). \tag{2.155}$$

Combining this equation with the pressure-matching equation (2.150) we can obtain an explicit expression for $\mathbf{a}^{j+1}$ and $\mathbf{b}^{j+1}$:

$$\begin{bmatrix} \mathbf{a}^{j+1} \\ \mathbf{b}^{j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^j & \mathbf{R}_2^j \\ \mathbf{R}_3^j & \mathbf{R}_4^j \end{bmatrix} \begin{bmatrix} \mathbf{a}^j \\ \mathbf{b}^j \end{bmatrix}, \tag{2.156}$$

where,

$$\begin{aligned}
\mathbf{R}_1^j &= \frac{1}{2}\left(\tilde{\mathbf{C}}^j + \hat{\mathbf{C}}^j\right)\mathbf{H}_1^j \\
\mathbf{R}_2^j &= \frac{1}{2}\left(\tilde{\mathbf{C}}^j - \hat{\mathbf{C}}^j\right)\mathbf{H}_2^j \\
\mathbf{R}_3^j &= \frac{1}{2}\left(\tilde{\mathbf{C}}^j - \hat{\mathbf{C}}^j\right)\mathbf{H}_1^j \\
\mathbf{R}_4^j &= \frac{1}{2}\left(\tilde{\mathbf{C}}^j + \hat{\mathbf{C}}^j\right)\mathbf{H}_2^j.
\end{aligned} \tag{2.157}$$

Finally, we need to include the boundary condition at $r = 0$ and a radiation condition as $r \to \infty$. The latter is imposed by requiring that $b_m^N = 0$, for $m = 1, \ldots, M$. The appropriate condition at $r = 0$ can be shown to be:

$$a_m^1 = \frac{i}{4\rho(z_s)} Z_m(z_s) H_0^{(1)}(k_m^1 r_1) + b_m^1 \frac{H_0^{(1)}(k_m^1 r_1)}{H_0^{(2)}(k_m^1 r_1)}, \qquad m = 1, \ldots, M. \tag{2.158}$$

Collecting all of these equations together we obtain a block matrix problem of the following form:

$$\begin{bmatrix}
\mathbf{I} & -\mathbf{D} & 0 & & & & & \\
\mathbf{R}_1^1 & \mathbf{R}_2^1 & \mathbf{I} & 0 & & & & \\
\mathbf{R}_3^1 & \mathbf{R}_4^1 & 0 & \mathbf{I} & & & & \\
 & \ddots & \ddots & \ddots & \ddots & & & \\
 & & \mathbf{R}_1^{N-2} & \mathbf{R}_2^{N-2} & \mathbf{I} & 0 & & \\
 & & \mathbf{R}_3^{N-2} & \mathbf{R}_4^{N-2} & 0 & \mathbf{I} & & \\
 & & & & \mathbf{R}_1^{N-1} & \mathbf{R}_2^{N-1} & \mathbf{I} & \\
 & & & & \mathbf{R}_3^{N-1} & \mathbf{R}_4^{N-1} & 0 &
\end{bmatrix}
\begin{bmatrix} \mathbf{a}^1 \\ \mathbf{b}^1 \\ \mathbf{a}^2 \\ \mathbf{b}^2 \\ \vdots \\ \mathbf{a}^{N-1} \\ \mathbf{b}^{N-1} \\ \mathbf{a}^N \end{bmatrix}
=
\begin{bmatrix} \mathbf{s} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{2.159}$$

where $\mathbf{D}$ is the diagonal matrix with entries

$$d_{ii} = \frac{H_0^{(1)}(k_i^1 r_1)}{H_0^{(2)}(k_i^1 r_1)}, \tag{2.160}$$

and $\mathbf{s}$ is the column vector with entries:

$$s_m = \frac{i}{4\rho(z_s)} Z_m(z_s) H_0^{(1)}(k_m^1 r_1). \tag{2.161}$$

Computationally, this approach requires the solution of a whole family of normal mode problems, one for each range segment, followed by the solution of a large banded, block linear system. The range segments are frequently required to be less than a wavelength which leads to extremely long run times.

Finally, we should also mention that it is possible to formulate the coupled-mode problem in a differential form. In fact, we shall follow this approach in a subsequent section since it leads more naturally to the adiabatic mode approximation. This development, in terms of piecewise range-independent sections, may then be viewed as simply one way of solving the coupled-differential equations in range.

2.8.2 One-way Coupled Modes

The full two-way coupled mode formulation allows for interactions between each segment in range and as a result leads to a global problem rather than a marching type of solution provided by, for instance, the parabolic equation. Computation time can be reduced by neglecting these multiple interactions, usually with only a minor degradation in accuracy.

An efficient marching implementation of coupled modes can be done in several ways with different degrees of accuracy. This is discussed in detail in Ref. [80]. A good compromise between accuracy and complexity is provided by the *single-scatter* formulation which treats each interface in range as an independent process thus neglecting the higher-order multiple-scattering terms. To derive this form we begin with the matching condition for the jth interface given in Eq. (2.156):

$$\begin{bmatrix} \mathbf{a}^{j+1} \\ \mathbf{b}^{j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{R}_3 & \mathbf{R}_4 \end{bmatrix} \begin{bmatrix} \mathbf{a}^{j} \\ \mathbf{b}^{j} \end{bmatrix}. \tag{2.162}$$

For the single-scatter approximation, the incoming wave in the left segment is assumed to be given, and we require that the solution is purely outgoing in the right segment, i.e. $\mathbf{b}^{j+1} = \mathbf{0}$. Solving for the backscattered amplitudes, $\mathbf{b}^{j}$, we find:

$$\mathbf{b}^{j} = -\mathbf{R}_4^{-1}\mathbf{R}_3\mathbf{a}^{j}. \tag{2.163}$$

Therefore, the forward scattered amplitudes, $\mathbf{a}^{j+1}$, are given by

$$\mathbf{a}^{j+1} = \left(\mathbf{R}_1 - \mathbf{R}_2\mathbf{R}_4^{-1}\mathbf{R}_3\right)\mathbf{a}^{j}, \tag{2.164}$$

which is an explicit equation for the forward scattered field. The field in any given segment can then be computed by summing the terms in the modal sum representing the forward scattered field.

In practice, an *approximate single-scatter* solution works nearly as well. This solution is obtained by neglecting lower-order terms in the single-scatter recursion:

$$\mathbf{a}^{j-1} = \mathbf{R}_1\mathbf{a}^{j}. \tag{2.165}$$

It can be shown that the matrix $\mathbf{R}_1$ is an arithmetic mean of coupling matrices based on pressure matching and velocity matching.

2.8.3 The Adiabatic Approximation

The one-way coupled mode approach discussed in the previous section provides a great speed-up in execution time but in many cases still remains too time consuming. For this reason a further approximation is often invoked in which one neglects the cross-coupling terms which allow energy from one mode to transfer into other modes. Instead, one assumes that in going from one range to the next the modes will couple *adiabatically*, that is, without any transfer of energy to other modes. This approximation was introduced to ocean acoustics problems by Pierce [54] based on

analogous results for the Schrödinger equation. Our derivation follows Pierce, however we note that a somewhat more formal derivation is given by Weinberg and Burridge in Ref. [55].

To derive this approximation, we return to the Helmholtz equation in two-dimensions:

$$\frac{\rho}{r}\frac{\partial}{\partial r}\left(\frac{r}{\rho}\frac{\partial p}{\partial r}\right)+\rho\frac{\partial}{\partial z}\left(\frac{1}{\rho}\frac{\partial p}{\partial z}\right)+\frac{\omega^2}{c^2(r,z)}p=\frac{-\delta(z-z_s)\delta(r)}{2\pi r}. \tag{2.166}$$

Since the modes form a complete set, we can represent the solution at any range as a sum of local modes. We therefore seek a solution of the range-dependent problem in the form

$$p(r,z)=\sum_m R_m(r)Z_m(z,r), \tag{2.167}$$

where, $Z_m(r,z)$ are the local modes defined by

$$\rho(r,z)\left[\frac{1}{\rho(r,z)}Z_m'(z,r)\right]'+\left(\frac{\omega^2}{c^2(r,z)}-k_m^2(r)\right)Z_m(z,r)=0. \tag{2.168}$$

and primes denote differentiation with respect to z. Thus, at any range r, $Z_m(r,z)$ is found by solving the depth-separated modal equation with the environmental properties at that range. Substituting in the Helmholtz equation yields:

$$\sum_m\frac{\rho}{r}\frac{\partial}{\partial r}\left(\frac{r}{\rho}\frac{\partial(R_mZ_m)}{\partial r}\right)-\sum_m k_m^2(r)R_mZ_m=\frac{-\delta(z-z_s)\delta(r)}{2\pi r}. \tag{2.169}$$

where we have used Eq. (2.168) to eliminate the z-derivatives. Rearranging terms leads to,

$$\sum_m\left[\frac{\rho}{r}\frac{\partial}{\partial r}\left(\frac{r}{\rho}\frac{\partial R_m}{\partial r}\right)Z_m+2\frac{\partial R_m}{\partial r}\frac{\partial Z_m}{\partial r}+\frac{\rho}{r}\frac{\partial}{\partial r}\left(\frac{r}{\rho}\frac{\partial Z_m}{\partial r}\right)R_m\right]$$
$$-\sum_m k_m^2(r)R_mZ_m=\frac{-\delta(z-z_s)\delta(r)}{2\pi r}. \tag{2.170}$$

For simplicity we shall now assume that ρ is independent of r. Then we can apply the operator

$$\int(\cdot)\frac{Z_l(z,r)}{\rho}dz. \tag{2.171}$$

and because of the orthogonality property many of the terms in the sum will disappear. The result is:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial R_l}{\partial r}\right)+\sum_m 2B_{lm}\frac{\partial R_m}{\partial r}+\sum_m A_{lm}R_m+k_l^2(r)R_l=\frac{-Z_l(z_s)\delta(r)}{2\pi r}. \tag{2.172}$$

where,

$$\begin{aligned}A_{lm}&=\int\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial Z_m}{\partial r}\right)\frac{Z_l}{\rho}dz,\\ B_{lm}&=\int\frac{\partial Z_m}{\partial r}\frac{Z_l}{\rho}dz.\end{aligned} \tag{2.173}$$

Note that $B_{lm} = -B_{ml}$, since differentiating

$$\int \frac{Z_m(z)Z_l(z)}{\rho(z)}\,dz = \delta_{lm}, \tag{2.174}$$

gives

$$\int \frac{\partial Z_m(z)}{\partial r}\frac{Z_l(z)}{\rho(z)}\,dz + \int \frac{Z_m(z)}{\rho(z)}\frac{\partial Z_l(z)}{\partial r}\,dz = 0. \tag{2.175}$$

Equation (2.172) is a statement of coupled modes written for the case of continuous variation of sound speed. It can be solved directly by, for instance, finite differences. The adiabatic approximation can now be stated simply as the assumption that the coupling matrices B_{lm} and A_{lm} are negligible. (Some authors retain the diagonal terms A_{nn} which alters the results slightly.) This yields the set of decoupled equations

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial R_l}{\partial r}\right) - k_l^2(r)R_l = \frac{-Z_l(z_s)\delta(r)}{2\pi r}. \tag{2.176}$$

The WKB approximation then yields the solution:

$$R_l(r) \approx A\frac{e^{i\int_0^r k_l(s)\,ds}}{\sqrt{k_l(r)}}. \tag{2.177}$$

The value of A is found by requiring that the WKB solution must match our normal solution, Eq. (2.16), when the problem is range-independent. Thus,

$$A = \frac{i}{\sqrt{8\pi r}}e^{-i\pi/4}Z_l(z_s). \tag{2.178}$$

Substituting this results back into Eq. (2.167) we obtain the final result:

$$p(r,z) \approx \frac{i}{\sqrt{8\pi r}}e^{-i\pi/4}\sum_{m=1}^{\infty} Z_m(z_s)Z_m(z,r)\frac{e^{i\int_0^r k_m(s)\,ds}}{\sqrt{k_m(r)}}. \tag{2.179}$$

In practice, the eigenfunctions and eigenvalues are normally calculated at a discrete set of ranges: values at intermediate ranges are then calculated by linear interpolation. Note that the adiabatic form is sensitive to the polarity of the modes: that is, if you flip the sign of $Z_m(z,r)$ at some particular range, the computed pressure is changed. Therefore, care must be taken that the modes are polarized in a consistent fashion in range. In KRAKEN the modes are polarized so that they assume a positive value at the turning point nearest to the surface.

2.8.4 Example: A Warm-Core Eddy

Range-dependence can occur as both bathymetric variation (due to seamounts, continental slopes) and variation in material properties (due to oceanographic features such as fronts and eddies or to changes in bottom type). We consider a flat-bottom problem involving an eddy with a source located roughly above the center of the

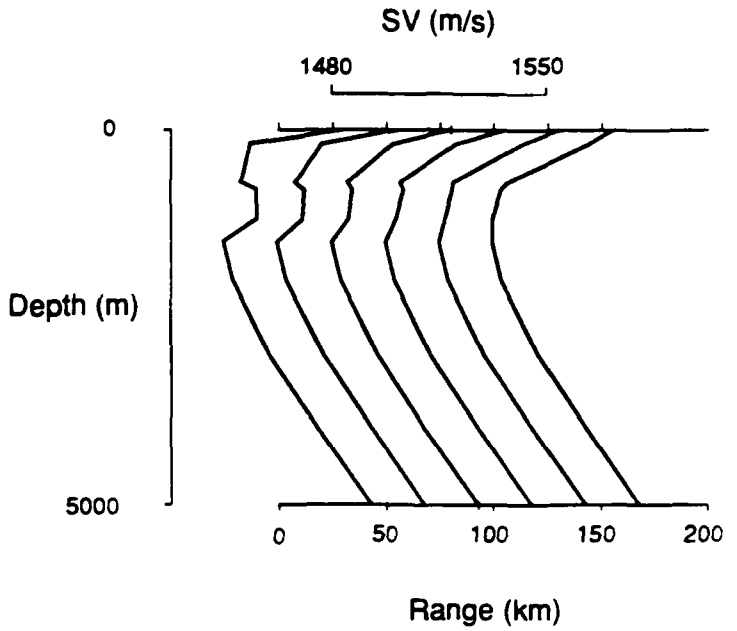


Figure 2.14: Sound speed profiles taken through the eddy.

eddy. The actual sound speed profiles are shown in Fig. 2.14. The warm-core eddy which is centered at roughly 1000 m depth and at zero range, shows up as a zone of increased sound speed.

Figure 2.15 shows plots of the transmission loss obtained with a) range-independent normal modes using the first SSP throughout, b) coupled mode theory and c) adiabatic normal modes. The source frequency is 50 Hz and the source depth is 300 m. The range-independent calculation in Fig. 2.15(a) shows deep-cycling convergence-zone paths. A band of energy is also seen to propagate in the duct that is roughly centered at the source depth in the first profile.

The one-way coupled-mode calculation in Fig. 2.15(b) shows that as the duct disappears in range the energy passes into the main SOFAR duct. The result is a great increase in transmission loss for a receiver located at, for instance, 100 m depth. Thus, in this particular case a range-independent calculation would almost certainly be considered inadequate.

The faster adiabatic calculation shown in Fig. 2.15(c) provides an intermediate result in terms of accuracy: it correctly shows the transition of energy from the near surface duct into the main SOFAR duct but fails to reproduce the details of the pattern. Whether this result would be considered adequate depends on the application.

The adiabatic approximation provides accurate predictions when the range-dependence is sufficiently weak. What constitutes "weak" range-dependence? This is a question which has been addressed in numerous papers but is difficult to answer in any general sense. In some cases the adiabatic approximation provides very

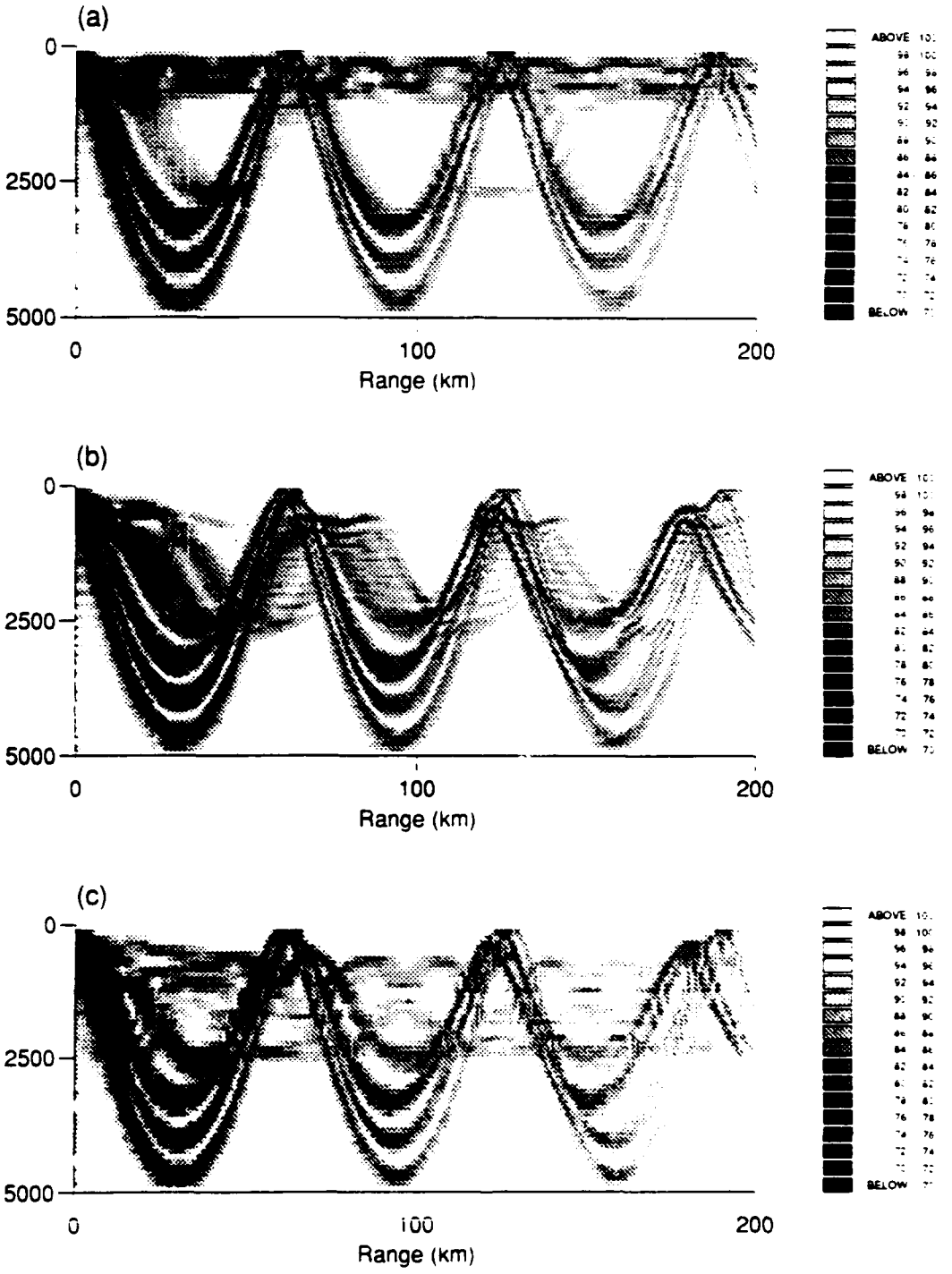


Figure 2.15: Transmission loss for the eddy problem using (a) range-independent, (b) coupled and (c) adiabatic mode theory.

poor results. An interesting example of such a problem is obtained by introducing a seamount into our eddy problem. If the seamount is placed in the shadow zone between 100 and 150 km then the convergence zone paths will pass over it unaffected. If the seamount is placed in the ensonified area centered at 90 km then the convergence zone paths will reflect off the seamount and be displaced in range. However, past the seamount the adiabatic formula depends only on the integral of the wavenumbers and is therefore *insensitive* to the location of the seamount. Despite these problems, the adiabatic approximation is frequently very useful.

2.9 Normal Modes for 3-D Varying Environments

As with all of the range-dependent 2-D models, a direct extension to 3-D problems is possible by simply running the 2-D models repeatedly along a number of different bearings. Along each bearing one then uses the sound speed profile and bathymetry for that track. Combining these results along numerous bearings allows one to build-up a three-dimensional image of the acoustic field.

As in the 2-D problem it is convenient to calculate the mode sets on a coarse grid and calculate intermediate values by interpolation. As an example, we consider a scenario in the North Atlantic which encompasses a segment of the Gulf Stream. The environment is sampled at a number of different points in the xy-plane and modes are calculated at each of these points. The nodes are then used to construct a triangulation of the environment as shown in Fig. 2.16. The position of the Gulf stream, two eddies that have spun-off from it, and other features of the environment are echoed in the triangulation.

The nodes of the triangles may be arbitrarily located; however, we have found it convenient to distribute them along isobaths. This enables extremely complicated bottom profiles to be treated without the burden of computing sets of modes at many different nodes— modes computed at one point on an isobath are unchanged (assuming no sound speed change) in going to other points on the isobath. Regular grids are also suitable: they are easier to set-up but may require more nodal points to produce an acceptable sampling of the environment.

Once the modes are calculated and stored, 3-D acoustic images can be computed extremely rapidly by using the adiabatic formula in Eq. (2.179) along a fan of radials emanating from the source. Where the adiabatic formula requires modes in a triangle, they are computed by bilinear interpolation. (A more complete discussion is given in Ref. [17].)

An example of this type of calculation is given in Fig. 2.17 where we have plotted transmission loss in the xy-plane at a constant receiver depth of 400 m. The source depth and frequency for this calculation are 400 m and 50 Hz. Note how the Gulf stream casts an acoustic shadow behind it. (In fact, the energy is really just redirected to other depths.) The eddies in turn produce visible perturbations in the transmission loss field.

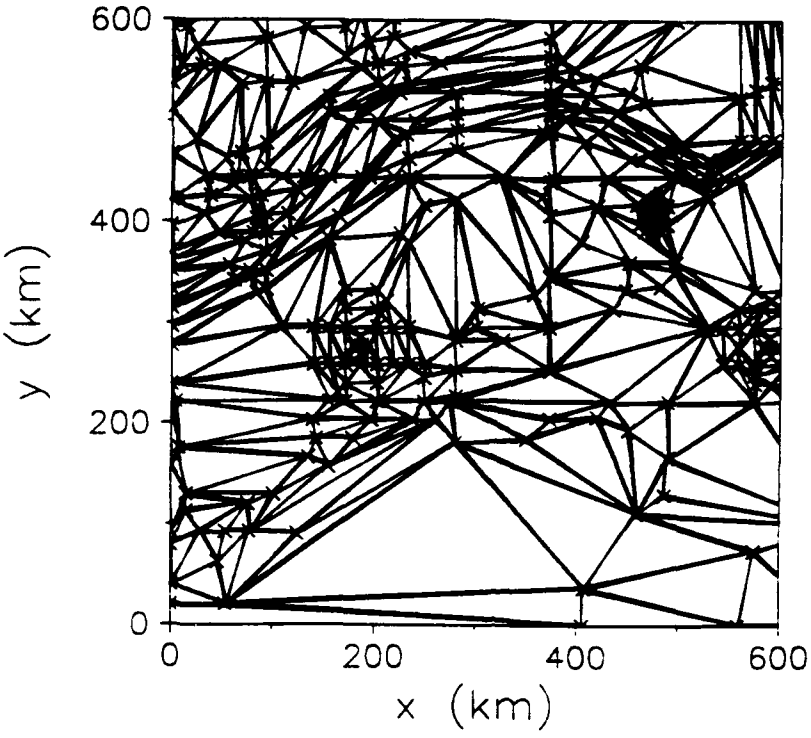


Figure 2.16: Triangulation for the Gulf Stream problem.

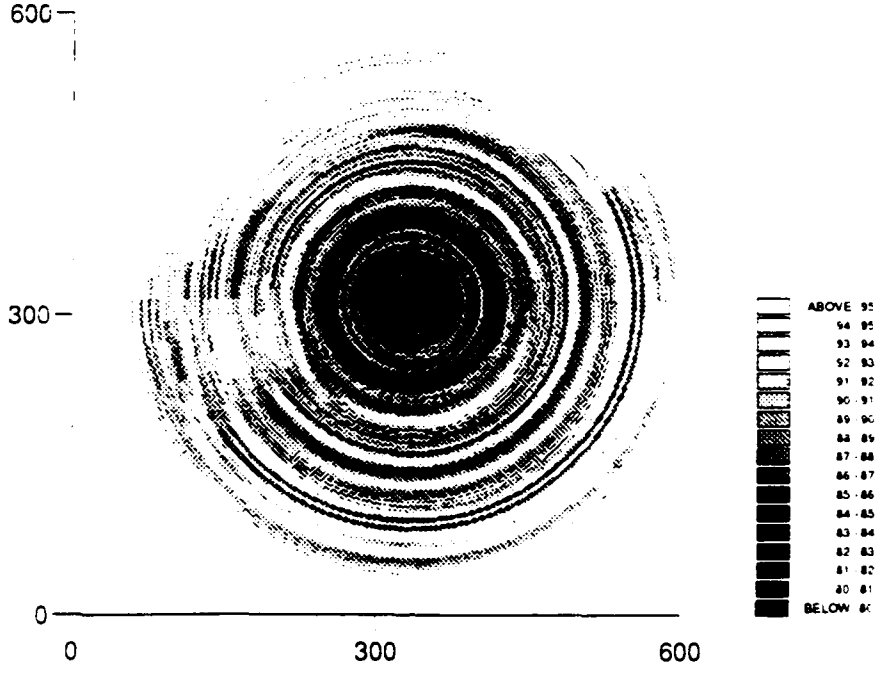


Figure 2.17: Transmission loss for the Gulf Stream problem.

2.9.1 Horizontal Refraction Equations

The range-dependent 2-D models assume azimuthally-symmetric environments and we violate this assumption in applying the model with different profiles for each bearing. In practice, this approximation is generally adequate and indeed is normally implicit in a 2-D run of a range-dependent model. That is, we normally use environmental information on a bearing-line between source and receiver, not intending the slice to define an azimuthally-symmetric environment.

In some cases, effects of horizontal refraction must be included. In principle, this is easily done. The derivation follows the pattern developed in the previous section for adiabatic modes in two-dimensions. We start with the Helmholtz equation in three-dimensions:

$$\rho\nabla\cdot\left(\frac{1}{\rho}\nabla p\right)+\frac{\omega^2}{c^2(x,y,z)}p=-\delta(x)\delta(y)\delta(z-z_s), \tag{2.180}$$

or, written out in full,

$$\rho\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial p}{\partial z}\right)+\rho\frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial p}{\partial z}\right)+\rho\frac{\partial}{\partial z}\left(\frac{1}{\rho}\frac{\partial p}{\partial z}\right)+\frac{\omega^2}{c^2(x,y,z)}p = -\delta(x)\delta(y)\delta(z-z_s).$$

We next seek a solution of the form

$$p(x,y,z)=\sum_m \phi_m(x,y)Z_m(x,y,z), \tag{2.181}$$

where $Z_m(x,y,z)$ are the local modes. Substituting in the Helmholtz equation and applying the operator

$$\int(\cdot)\frac{Z_l(x,y.z)}{\rho}dz, \tag{2.182}$$

gives:

$$\frac{\partial^2\phi_l}{\partial x^2}+\frac{\partial^2\phi_l}{\partial y^2}+k_l^2(x,y)\phi_l+\sum_m A_{lm}\phi_m + \sum_m 2B_{lm}\frac{\partial\phi_m}{\partial x}+\sum_m 2C_{lm}\frac{\partial\phi_m}{\partial y}=-\delta(x)\delta(y)\delta(z-z_s), \tag{2.183}$$

where,

$$\begin{aligned} A_{lm} &= \int\left(\frac{\partial^2}{\partial x^2}-\frac{\partial^2}{\partial y^2}\right)Z_m\frac{Z_l}{\rho}dz. \\ B_{lm} &= -B_{ml}=\int\frac{\partial Z_m}{\partial x}\frac{Z_l}{\rho}dz, \\ C_{lm} &= -C_{ml}=\int\frac{\partial Z_m}{\partial y}\frac{Z_l}{\rho}dz. \end{aligned} \tag{2.184}$$

The adiabatic approximation can then be obtained by neglecting the contributions of the coupling matrices A, B and C. This yields the horizontal refraction equation:

$$\frac{\partial^2 \phi_l}{\partial x^2} + \frac{\partial^2 \phi_l}{\partial y^2} - k_l^2(x,y)\phi_l = -Z_l(z_s)\delta(x)\delta(y). \tag{2.185}$$

Using the local normal modes, we have eliminated the z-dimension from the problem and obtained a new Helmholtz equation, but now in the lateral coordinates x and y. The effective index of refraction is given by the horizontal wavenumber $k_l(x,y)$ so that every mode generates a corresponding Helmholtz equation. Such 2-D Helmholtz equations can be solved using normal modes, PE, ray or spectral integral methods. Weinberg and Burridge[55] present results using ray theory to solve the horizontal refraction equations.

Note that the horizontal refraction equation leads to the usual results for the cases of stratified range-independent or range-dependent but cylindrically symmetric profiles. For instance, if we rewrite the horizontal refraction equation in cylindrical coordinates and assume a cylindrically symmetric sound speed profile, we obtain,

$$\frac{1}{r}(r\phi_l')' + k_l(r)\phi_l = \varphi_l(z_s;0)\frac{\delta(r)}{r}. \tag{2.186}$$

The WKB approximation to the solution is then,

$$\phi_l(r) = Z_l(z_s;0)\frac{e^{-i\int_0^r k_l(\tilde{r})d\tilde{r}}}{\sqrt{k_l(r)r}}. \tag{2.187}$$

Using this result in Eq. (2.181) yields the usual adiabatic mode result of Eq. (2.179).

In KRAKEN the horizontal refraction equations are solved using Gaussian beams which refract in the horizontal plane. As we have just seen, the "refractive medium" for each set of Gaussian beams is constructed from the phase velocity field corresponding to that particular mode. The field due to an individual beam involved a Gaussian decay in the horizontal plane and a depth dependence determined by the local mode.

The Gaussian beam solution to the horizontal refraction equation is straightforward. As discussed in references [59,60,61] the method begins by tracing a fan of beams originating from the source. The beams are defined by a central ray (which obeys the usual ray equations), the beam radius, $L(s)$ and the curvature, $K(s)$, which are functions of arclength along the ray. The beams are Gaussian in that at any point on the central ray of an individual beam the intensity falls-off in a Gaussian form as a function of normal distance from the central ray. The final step is to sum up the contributions of all of the beams to compute the solution.

In more detail, the process is as follows. We begin by solving for the central rays of the beams which satisfy ray equations:

$$\begin{aligned} \frac{dx}{ds} &= c_l\xi(s), \\ \frac{d\xi}{ds} &= -\frac{1}{c_l^2}\frac{dc_l}{dx}. \end{aligned}$$

$$\begin{aligned} \frac{dy}{ds} &= c_l\eta(s), \\ \frac{d\eta}{ds} &= -\frac{1}{c_l^2}\frac{dc_l}{dy}, \end{aligned} \tag{2.188}$$

where, s denotes arclength along the ray and $c_l(x,y)$ is the phase speed of the jth mode. A beam is then constructed about each ray using ray-centered coordinates, (s,n) where n denotes the normal distance from the ray. The result is,

$$g_l(s,n) = A\sqrt{\frac{c(s)}{q(s)}}\exp\left\{-i\omega\left[\tau(s)+\frac{p(s)}{2q(s)}n^2\right]\right\} \tag{2.189}$$

where the complex quantities $p(s)$ and $q(s)$ are obtained by integrating an auxiliary set of differential equations,

$$\frac{dq}{ds} = c(s)p(s), \tag{2.190}$$

$$\frac{dp}{ds} = -\frac{c_{nn}(s)}{c^2(s)}q(s). \tag{2.191}$$

where the subcript n in c_{nn} indicates the derivative of the sound speed in a direction normal to the central ray.

The term $\tau(s)$ in Eq. (2.189) is the phase delay or travel time which satisfies,

$$\frac{d\tau}{ds} = \frac{1}{c(s)}. \tag{2.192}$$

It is convenient to think of the $p-q$ functions as defining the evolution of the beam in terms of its radius and curvature. Consider,

$$L(s) = \sqrt{-2\omega\mathrm{Im}\left\{\frac{p(s)}{q(s)}\right\}}, \tag{2.193}$$

$$K(s) = -c(s)\mathrm{Re}\left\{\frac{p(s)}{q(s)}\right\}. \tag{2.194}$$

Clearly, $L(s)$ characterizes the beam radius in terms of the normal distance at which the beam decays by $1/e$. Furthermore, $K(s)$ is a representation of the number of phase fronts crossed as one travels normal to the central ray and is a measure of curvature of the beam.

To summarize, the beam is defined in terms of $(x(s),y(s))$, $(\xi(s),\eta(s))$, $(p(s),q(s))$, and $\tau(s)$ representing respectively the central ray of the beam. the tangent to the central ray, the beam width and curvature, and the travel time. All of these functions are obtained by integrating a set of ordinary differential equations. To complete the definition we must provide initial conditions. For the central ray these are simply,

$$\begin{aligned} x(0) &= x_s, \\ y(0) &= y_s, \\ \xi(0) &= \frac{\cos\alpha}{c(0)}, \\ \eta(0) &= \frac{\sin\alpha}{c(0)}. \end{aligned} \tag{2.195}$$

where (x_s, y_s) is the source coordinate and α is the take-off angle of the central ray.

The optimal choice of initial conditions for the $p - q$ equations is a matter of current research. The two extremes of infinitely wide and infinitely narrow beams may be favored in cases with strong variation of wave speeds or when head waves are generated. Fortunately, the horizontal refraction equations are especially benign. In contrast to the usual $r - z$ ocean acoustic problem, there are no boundaries to worry about. Secondly, the effective sound speed is so gradually varying that in may cases there is no clear need to even try to account for it. We have obtained good results using the analytic result for an isovelocity medium and selecting the initial beam width to minimize the beam width at the receiver. The beam curvature is chosen to be zero at the origin.

The final step is to synthesize the field by summing up the Gaussian beams. Thus, we seek a solution,

$$u(x, y, z) = \int_0^{2\pi} A(\alpha) g_l(s, n) d\alpha. \tag{2.196}$$

where $g_l(s, n)$ is the lth beam. Following the standard procedure, we construct the expansion by the method of canonical problems, i.e., we require that the beam expansion be reasonable for a homogeneous medium. In a homogeneous medium, the equations which define the beam may be solved analytically. In addition analytical representations for the receiver coordinate (x_r, y_r) in terms of the ray-centered coordinates (s, n) may also be obtained without fanfare. Finally, one applies the saddle point method to compute the high-frequency asymptotic expansion of the integral. The result is,

$$u(x, y) = \frac{A(\alpha_0)}{\sqrt{4\pi}} e^{-ik_l\sqrt{x^2+y^2}}. \tag{2.197}$$

On the other hand, the exact solution for a single-mode propagating in a stratified medium is

$$u(x, y) = Z_l(z_s) e^{-ik_l\sqrt{x^2+y^2}}. \tag{2.198}$$

Evidently the forms match if the weighting for each beam, $A(\alpha)$ is $Z_l(z_s)/\sqrt{4\pi}$. Putting all these things together, we obtain

$$p(x, y, z) = \sum_{n=1}^{m} o_l(x, y) Z_l(z). \tag{2.199}$$

where

$$o_l(x, y, z) = \int g_l(s, n) d\alpha. \tag{2.200}$$

and

$$g_l(s, n) = A\sqrt{\frac{c(s)}{q(s)}} \exp\left\{-i\omega\left[\tau(s) - \frac{p(s)}{2q(s)}n^2\right]\right\}. \tag{2.201}$$

The procedure then is to begin a loop over each mode. For each mode one has a Helmholtz equation with that mode's phase speed acting like the usual ocean sound speed. Each such horizontal refraction equation is solved by looping over an

azimuthal take-off angle and each azimuthal take-off angle leads to a beam which propagates out from the source. As each beam is traced out its contribution to the field is summed up. This procedure is repeated for each mode to obtain the complete field representation.

Chapter 3

Numerical Solution of the Modal Equation

There are many ways of numerically treating the modal equation. Generally, core storage is not a problem so that the algorithms can be impartially compared by setting an accuracy threshold and seeing which method requires the least execution time to meet that criterion on a typical set of test problems. The algorithm used in KRAKEN was chosen by just such a fly-off of numerous different algorithms. This included 3- and 5-point difference schemes, Numerov's method, iterated defect corrections and layer methods. Richardson extrapolation was also studied using polynomial and rational (Padé) approximations with the various algorithms. Several root-finders were also compared including newton, secant, and bisection schemes. The algorithm described below was the most rapid across the suite of test problems; however, it should be noted that the optimal algorithm is partly a function of the types of problems forming the test ensemble.

In the following sections we shall describe the algorithm which emerged as the fastest technique. Sections 3.1–3.3 describe the finite-difference discretization which leads to an algebraic eigenvalue problem (EVP). The EVP is solved using what is sometimes called the determinant search method as described in Sect. 3.4. Section 3.5 describes the treatment of elastic media and Sect. 3.6 discusses the use of Richardson extrapolation which provides an adaptive technique for controlling the error as well as an efficient means of obtaining high-accuracy.

3.1 Finite-Difference Discretization

As illustrated in Fig. 3.1 we divide the interval $0 < z < D$ into N equal intervals to construct a mesh of equally spaced points

$$z_j = jh, \qquad j = 0, \ldots, N \tag{3.1}$$

where h is the mesh width given by $h = D/N$. Furthermore, we shall use the notation $Z_j = Z(z_j)$. The number N should be chosen large enough so that the modes are adequately sampled; usually 10 points per wavelength are sufficient.

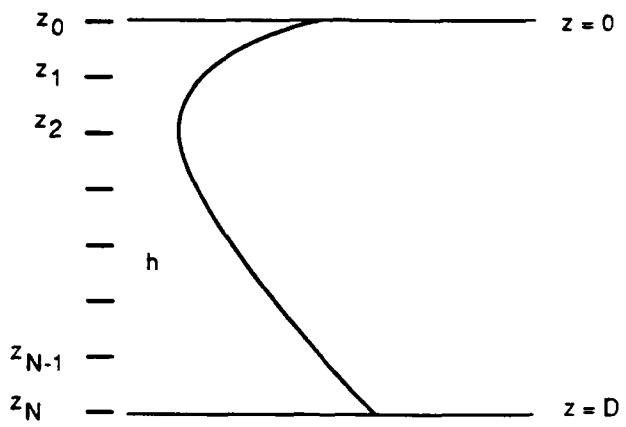


Figure 3.1: Finite-difference mesh.

We shall assume for the moment that the density is constant, yielding the modal problem

$$Z''(z) + \left(\frac{\omega^2}{c^2(z)} - k^2\right) Z(z) = 0, \tag{3.2}$$

where the primes denote differentiation with respect to z. Following a standard procedure for deriving finite difference equations, we use the Taylor series expansion to obtain

$$Z_{j+1} = Z_j + Z_j' h + Z_j'' \frac{h^2}{2!} + Z_j^{(3)} \frac{h^3}{3!} - \cdots \tag{3.3}$$

Rearranging terms, we obtain a *forward-difference* approximation for the first derivative,

$$Z_j' = \frac{Z_{j+1} - Z_j}{h} - Z_j'' \frac{h}{2} - \cdots. \tag{3.4}$$

An $O(h)$ approximation to the first derivative is therefore,

$$Z_j' \approx \frac{Z_{j+1} - Z_j}{h}. \tag{3.5}$$

An improved approximation is obtained by using the governing equation Eq. (3.2) to evaluate the first term in the forward difference approximation. That is, we substitute

$$Z''(z) = -\left(\frac{\omega^2}{c^2(z)} - k^2\right) Z(z). \tag{3.6}$$

This yields the $O(h^2)$ approximation

$$Z_j' \approx \frac{Z_{j-1} - Z_j}{h} - \left(\frac{\omega^2}{c^2(z_j)} - k^2\right) Z_j \frac{h}{2}. \tag{3.7}$$

Similarly, a *backward-difference* approximation is obtained starting with the Taylor series

$$Z_{j-1} = Z_j - Z_j' h + Z_j'' \frac{h^2}{2!} + Z_j^{(3)} \frac{h^3}{3!} - \cdots \tag{3.8}$$

yielding, the $O(h)$ approximation

$$Z_j' \approx \frac{Z_j - Z_{j-1}}{h}, \tag{3.9}$$

and the $O(h^2)$ approximation

$$Z_j' \approx \frac{Z_j - Z_{j-1}}{h} + \left(\frac{\omega^2}{c^2(z_j)} - k^2\right) Z_j \frac{h}{2}. \tag{3.10}$$

Finally, adding Eqs. (3.3) and (3.8) we obtain a *centered-difference* approximation to the second derivative:

$$Z_j'' = \frac{Z_{j-1} - 2Z_j + Z_{j+1}}{h^2} + O(h^2). \tag{3.11}$$

With these finite-difference approximations in hand, we can proceed to replace the derivatives in the continuous problem with discrete analogues. Let us recall the continuous problem:

$$\begin{aligned} Z''(z) - \left(\frac{\omega^2}{c^2(z)} - k^2\right) Z(z) &= 0, \\ f^T(k^2) Z(0) - \frac{g^T(k^2)}{\rho} Z'(0) &= 0, \\ f^B(k^2) Z(D) - \frac{g^B(k^2)}{\rho} Z'(D) &= 0. \end{aligned} \tag{3.12}$$

Using the centered, forward and backward difference approximations for the ODE, the top and bottom boundary conditions we obtain:

$$\begin{aligned} Z_{j-1} - \left[-2 - h^2\left(\frac{\omega^2}{c^2(z_j)} - k^2\right)\right] Z_j - Z_{j-1} &= 0, \qquad j = 1, \ldots, N-1 \\ \frac{f^T}{g^T} Z_0 + \frac{1}{\rho}\left[\frac{Z_1 - Z_0}{h} - \left(\frac{\omega^2}{c^2(0)} - k^2\right) Z_0 \frac{h}{2}\right] &= 0, \\ \frac{f^B}{g^B} Z_N - \frac{1}{\rho}\left[\frac{Z_N - Z_{N-1}}{h} - \left(\frac{\omega^2}{c^2(D)} - k^2\right) Z_N \frac{h}{2}\right] &= 0. \end{aligned} \tag{3.13}$$

We next write the first of these equations as

$$\frac{1}{h\rho}Z_{j-1} + \frac{-2 + h^2\left[\omega^2/c^2(z_j) - k^2\right]}{h\rho}Z_j - \frac{1}{h\rho}Z_{j-1} = 0. \tag{3.14}$$

Then collecting the difference equations together we obtain an algebraic eigenvalue problem of the form:

$$\mathbf{A}(k^2)\mathbf{Z} = 0. \tag{3.15}$$

Here, $\mathbf{Z}$ is the vector with components $Z_0, Z_1, \ldots Z_N$. These components are approximations of the eigenfunctions of Eq. (3.2) evaluated at the mesh points. In addition, A is a symmetric tridiagonal matrix defined by

$$A = \begin{bmatrix} d_0 & e_1 & & & & & \\ e_1 & d_1 & e_2 & & & & \\ & e_2 & d_2 & e_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & e_{N-2} & d_{N-2} & e_{N-1} & \\ & & & & e_{N-1} & d_{N-1} & e_N \\ & & & & & e_N & d_N \end{bmatrix} \tag{3.16}$$

where the coefficients d_j and e_j are defined by

$$\begin{aligned} d_0 &= \frac{-2 + h^2\left[\omega^2/c^2(z_0) - k^2\right]}{2h\rho} - \frac{f^T(k^2)}{g^T(k^2)}, \\ d_j &= \frac{-2 + h^2\left[\omega^2/c^2(z_j) - k^2\right]}{h\rho}, \qquad j = 1, \ldots, N-1 \\ d_N &= \frac{-2 + h^2\left[\omega^2/c^2(z_N) - k^2\right]}{2h\rho} - \frac{f^B(k^2)}{g^B(k^2)}, \end{aligned} \tag{3.17}$$

and,

$$e_j = \frac{1}{h\rho}, \qquad j = 1, \ldots, N.$$

We have consciously introduced a scaling factor of $1/(h\rho)$ in every row here. For a constant-density problem with a single mesh width this would serve no purpose; however, later we shall consider multiple layers for which this scaling is the natural one.

Note that for a pressure-release surface the ratio f/g which appears in the boundary condition goes to infinity. In this case, Z_0 vanishes and we can simply delete the first line and column from the matrix eigenvalue problem. Furthermore, if the functions $f^{T,B}, g^{T,B}$ are independent of k (as happens for the pressure release surface and rigid bottom conditions) then the above problem is a standard algebraic eigenvalue problem and can be solved using standard routines. In general, only the lower-order modes will be sufficiently accurate: the higher order modes are undersampled by the finite-difference mesh. Thus, routines which are designed to extract a subset of the eigenvectors and eigenvalues are desired.

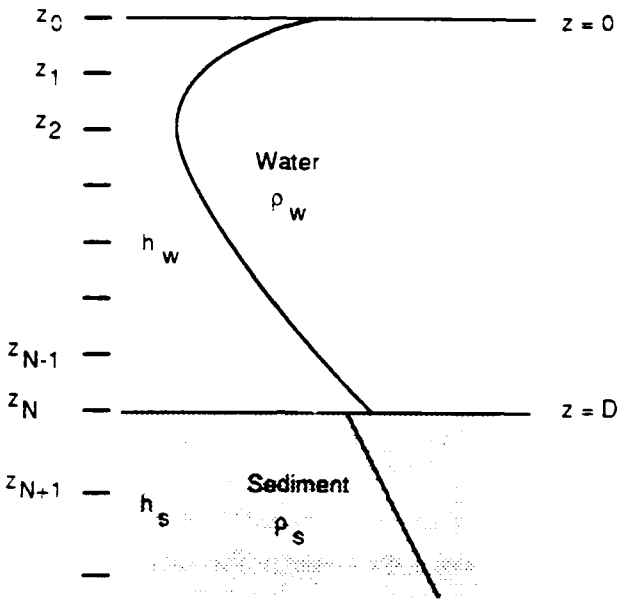


Figure 3.2: Finite-difference mesh for an interface.

The problem is a non-standard eigenvalue problem because the eigenvalue enters in a functional form through bottom boundary conditions. (For perfectly rigid or free boundary conditions the problem reduces to a classical algebraic eigenvalue problem.) We shall discuss the solution of the eigenvalue problem in Sect. 3.3.

3.2 Treatment of Interfaces

Frequently ocean acoustic problems involve discontinuities in the sound speed or density, for instance in passing from ocean to sediment. Such problems are treated by dividing the problem into layers such that within a layer these material properties vary smoothly. Within a layer the previous finite difference equations are applicable. At the interface one then derives a special condition to tie together the individual layers.

As an example, we consider a a single interface between two layers representing the water and the sediment. Within each layer we construct independent finite-difference meshes with grid spacing h_w and h_s as illustrated in Fig. 3.2. In the water, the finite-difference approximation to Eq. (3.2) is

$$Z_{j-1} - \left[-2 - h_w^2 \left(\frac{\omega^2}{c^2(z_j)} - k^2\right)\right] Z_j - Z_{j-1} = 0. \qquad j = 1, \ldots, N-1 \qquad (3.18)$$

and in the sediment the finite-difference approximation is

$$Z_{j-1} + \left[-2 + h_s^2\left(\frac{\omega^2}{c^2(z_j)} - k^2\right)\right] Z_j - Z_{j+1} = 0, \qquad j = N+1, \ldots \tag{3.19}$$

At the interface, the pressure must be continuous, a condition which is imposed implicitly by allowing for a unique value Z_N at the interface. We must also impose continuity of normal velocity, that is,

$$\frac{Z'(D)}{\rho_w} = \frac{Z'(D)}{\rho_s}, \tag{3.20}$$

where ρ_w and ρ_s denote the densities in the water and sediment respectively. This interface condition can then be approximated by,

$$\begin{aligned} &\left[\frac{Z_N - Z_{N-1}}{h_w} + \left(\frac{\omega^2}{c^2(D^-)} - k^2\right) Z_N \frac{h_w}{2}\right] \Big/ \rho_w \\ = &\left[\frac{Z_{N+1} - Z_N}{h_s} - \left(\frac{\omega^2}{c^2(D^+)} - k^2\right) Z_N \frac{h_s}{2}\right] \Big/ \rho_s, \end{aligned} \tag{3.21}$$

where we have used the backward difference formula for the water and the forward difference for the sediment. Furthermore, $c(D^{\pm})$ denotes the limiting value of the sound speed at the interface as approached from $z < D$ (D^-) and $z > D$ (D^+). Rearranging we obtain,

$$\begin{aligned} \frac{Z_{N-1}}{h_w \rho_w} &+ \frac{-Z_N + \left[\omega^2/c^2(D^-) - k^2\right] Z_N h_w^2/2}{h_w \rho_w} \\ &+ \frac{-Z_N + \left[\omega^2/c^2(D^-) - k^2\right] Z_N h_s^2/2}{h_s \rho_s} - \frac{Z_{N-1}}{h \rho_s} = 0. \end{aligned} \tag{3.22}$$

Note that if $h_w = h_s$, $\rho_w = \rho_s$ and $c(z)$ is continuous we obtain the same finite-difference formula given in Eq. (3.18) for a point not on an interface.

This process can obviously be repeated for every interface in the problem and, just as for the single layer case, leads to a symmetric matrix eigenvalue problem. Incidentally, the resulting problem is precisely equivalent to what one would obtain by using finite elements with hat-shaped basis functions and 'mass-lumping'.

3.3 Mode Normalization

Recall that the evaluation of the pressure requires the *normalized* modes. The normalization constant is

$$N_m = \int_0^D \frac{Z_m^2(z)}{\rho(z)} dz - \frac{1}{2k_m} \frac{d(f/g)^T}{dk}\bigg|_{k_m} Z_m^2(0) - \frac{1}{2k_m} \frac{d(f/g)^B}{dk}\bigg|_{k_m} Z_m^2(D). \tag{3.23}$$

The integral term can be evaluated by the trapezoidal rule. That is,

$$\int_0^D \frac{Z_m^2(z)}{\rho(z)} dz \approx \frac{1}{N}\left(\frac{1}{2}\omega_0 - \omega_1 - \omega_2 - \cdots - \omega_{N-1} - \frac{1}{2}\omega_N\right) \tag{3.24}$$

where

$$o_j = \frac{Z_j^2}{\rho(z_j)}. \tag{3.25}$$

In problems with density discontinuities, the trapezoidal rule is applied separately within each smooth region. Finally, the derivatives $d(f/g)^{T,B}/dk$ can be either evaluated analytically or by a simple centered-difference formula depending on their complexity. In order to accommodate a wide variety of boundary conditions without over-complicating the code, KRAKEN uses the difference approach.

3.4 Solving the Discretized Problem

The solution of the above matrix eigenvalue problem proceeds in two steps. First, the eigenvalues are found by applying a root finder to the characteristic equation.

$$\Delta(k^2) = 0 \tag{3.26}$$

where $\Delta(k^2)$ denotes the determinant of the matrix. Finally, once the eigenvalues are found, the eigenvectors are computed using inverse iteration. The inverse iteration is given by,

$$A(k_j^2)Z^{(l+1)} = Z^{(l)}. \tag{3.27}$$

The matrix A is (nearly) singular because k_j is (nearly) an eigenvalue of A. One may easily show that the effect of this iteration is to continually amplify the eigenvector component in the starting vector that corresponds to the eigenvalue k_j. Thus, the starting vector $Z^{(1)}$ may be picked fairly arbitrarily. After each iteration, the new iterate $Z^{(l+1)}$ is renormalized to avoid overflow. In addition, the growth of the iterates is used as a check for convergence. (Typically 2 iterations are sufficient.)

This inverse iteration technique is described in numerous texts on algebraic eigenvalue problems, see for instance, Wilkinson 62 . The usual development applies for this nonlinear eigenvalue problem with some minor restrictions. In particular, the inverse iteration will *not* be reliable in cases where one-sided shooting would be unstable for constructing the impedance conditions. There are two cases where this could be an issue: 1) when the 'internal reflection coefficient' option is used recklessly and 2) for certain elastic problems with internal ducts.

One of the most difficult aspects of normal mode computations is that of finding the roots of the characteristic equation. The fundamental difficulty is that many familiar root-finding algorithms (such as the secant method or Newton's method) will only converge to a particular root if an initial guess is provided which is sufficiently close. Unfortunately, even though the eigenvalues of a purely acoustic problem are guaranteed to be distinct, they can be very nearly degenerate. As a result, many existing mode codes provide accurate but *incomplete* mode sets.

Two root-finding techniques are used in the KRAKEN program which we shall describe next. The first is efficient and robust, in fact fool-proof, but is applicable only to purely acoustic problems, or acoustic problems with an elastic half-space. The second is less reliable but valid for problems with elastic layers.

3.4.1 Method I: Sturm Sequences

For a purely acoustic problem a bisection approach is generally the most attractive solution. For this fairly broad class of problems the method will calculate a particular subset of the modes in a predictable amount of time *without fail*. However, for more complicated problems, for instance problems involving elasticity or complex wave speeds this technique is not applicable.

In brief, the bisection algorithm relies on the interlace property of the Sturm-Liouville problem that the number of zeroes in a trial eigenfunction increases monotonically as k decreases. (Recall that the mth mode has m zeroes.) A discrete analogue of a trial eigenfunction is the vector S_j of principal minors of A. That is,

$$S_1 = 1, \tag{3.28}$$

$$S_j = d_j S_{j-1} - e_j^2 S_{j-2}. \tag{3.29}$$

For any fixed k^2 the number of sign changes in the sequence indicates the number of eigenvalues greater than that particular k^2. In addition, the function $S_N(k^2)$ is the characteristic equation, the roots of which are the eigenvalues. The first property is used to construct isolating intervals for the eigenvalues. Thus, for each eigenvalue one computes interval endpoints which contain exactly one eigenvalue. The second property is used by a root finder to refine the eigenvalues within their isolating intervals.

In more detail this process proceeds as follows. First we compute an upper bound on the eigenvalues using Gerschgorin's theorem. (Incidentally, Gerschgorin's theorem applied to the discrete problem yields the bound $k_{max} = \omega/c_{min}$ where c_{min} is the lowest sound speed in the problem.) This provides an upper bound for the mode search. There are an infinite number of modes so that a lower bound must be selected in some fashion. This bound is user specified, but if it exceeds the halfspace velocity in the problem, the bound is reduced to eliminate leaky modes from the problem.

Next, we take the midpoint of the interval and compute the number of modes to the right of the midpoint. Based on the number of zero-crossings in the trial eigenfunction one may decide whether the first eigenvalue lies to the left or to the right of the midpoint. The midpoint then becomes either a new lower bound or a new upper bound for the eigenvalue. This process of interval halving is repeated until the interval contains precisely one eigenvalue. With the isolating interval computed for the first mode, one then performs the same process for the second mode and so on. For subsequent modes an upper bound is available from the lower bound of the previous mode. In addition, information generated during the bisection for the first mode provides useful bounds for higher order modes. As a result the generation of all M of the isolating intervals typically requires little more than M bisection steps.

In the second stage these bracketing intervals are passed to sophisticated root finder (Brent's method [63]) which combines bisection, linear interpolation and inverse quadratic interpolation to provide an estimate of the eigenvalue. This latter root finder is *guaranteed* to converge given an isolating interval for the root. The

whole process is very efficient (compared to brute force linear searching) and robust (compared to techniques which rely on asymptotic estimates to provide initial guesses for a root finder).

In practice, the sequence S_j is replaced by p_j where,

$$p_j = S_j e_j^2 \tag{3.30}$$

The sequence p_j then satisfies the following recursion:

$$p_1 = 1 \tag{3.31}$$

$$p_j = b_j p_{j-1} - p_{j-2}. \tag{3.32}$$

where,

$$b_j = -2 + h_m^2 \left(\frac{\omega^2}{c^2(z_j)} - k^2 \right), \qquad i = 1, 2, \ldots, N, \tag{3.33}$$

The number of sign changes in p_j is the same as that for the original sequence, since e_j has no sign changes. However, p_j is well scaled and requires half as many floating-point operations to compute. Incidentally, this latter sequence is precisely equivalent to what one obtains in a simple one-sided shooting scheme. Instabilities often occur in this sequence which are removed by simply scaling it down as it gets too large. The scale factors are retained and passed separately to the root finder. Surprisingly, this instability does not in the least affect the accuracy of the eigenvalues. This is proven in Wilkinson's text [62].

We have glossed over a difficult aspect of this method. Namely, the information provided by the Sturm sequence (regarding the number of modes to the right of some trial eigenvalue) is only justified for purely acoustic problems with free or rigid boundaries. However, for homogeneous half-spaces or elastic half-spaces there exists a simple correction to the count[16,71]. For elastic or acousto-elastic problems with more than one elastic layer or elastic gradients there is no useful extension and we are forced to use the deflation method described next.

3.4.2 Method II: Deflation

The philosophy of the deflation method is to start above the first eigenvalue and use the secant method to find the first eigenvalue. Once that eigenvalue is located it is deflated, that is, divided out of the characteristic equation, and the process is repeated for each eigenvalue in turn. Interestingly, one may formally prove that the secant method will *always* converge to the first eigenvalue if started at a point above that first eigenvalue. (See Wilkinson, [62].) Naturally, there are some footnotes to this sweeping statement. The key one is that the characteristic function should be a polynomial. Some care is required to avoid violating this requirement. Secondly, the eigenvalues should all be real. In practice, the deflation procedure works very well for most realistic ocean acoustic problems— even for complex eigenvalues. However, if branch cuts are present, problems are not unlikely.

The deflation of previous eigenvalues is a trivial process. Instead of computing the characteristic function, $\Delta(k^2)$, one computes:

$$\tilde{\Delta}(k^2) = \frac{\Delta(k^2)}{\prod_j k^2 - k_j^2} \tag{3.34}$$

where k_j are the previously computed eigenvalues which are to be removed or deflated.

3.5 Elastic Media

Mixed acousto-elastic problems are treated in one of two different ways. The first approach is to simply incorporate all of the elastic effects into a boundary condition. The particular boundary condition was already written down in Eq. (2.81). The coefficients of that boundary condition require the solution of Eq. (2.79) which is accomplished using a simple explicit integrator. Specifically, we employ the modified midpoint method which for a first-order system $\mathbf{Y}' = \mathbf{f}(z, \mathbf{Y})$ takes the form:

$$\mathbf{y}_0 = \mathbf{y}(z_0) \tag{3.35}$$

$$\mathbf{y}_1 = \mathbf{y}_0 + h_m \mathbf{f}(z_j, \mathbf{y}_j), \qquad j = 1, \ldots, N_m \tag{3.36}$$

and,

$$\mathbf{Y}_N = \frac{\mathbf{Y}_{N-1} + 2\mathbf{Y}_N + \mathbf{Y}_{N+1}}{4}. \tag{3.37}$$

The integration is carried out in succession through each of the elastic layers to compute the coefficients of the impedance boundary condition. This approach is used in KRAKEN and KRAKENC.

The other approach, which is implemented in KRAKEL, is simply to apply finite differences directly to the stress-displacement equations (2.74). The result is a somewhat complicated 9-diagonal matrix whose characteristic is evaluated using a LINPACK routine (DGBDI).

3.6 Richardson Extrapolation

The above sections provide an essentially complete description of a numerical algorithm for the modal problem. The techniques described involve the simplest finite difference schemes and therefore lead to a relatively simple algorithm. However, as it stands, the method would be inefficient compared to certain methods based on higher-order difference schemes, e.g. Numerov's method. The simplicity of the basic method however can be retained while gaining great improvements in efficiency by using Richardson extrapolation. This is the technique used in KRAKEN and is described next.

It can be shown that the numerically derived eigenvalues vary as a function of the mesh width h, as follows:

$$k^2(h) = k_0^2 + b_2 h^2 + b_4 h^4 - \cdots \tag{3.38}$$

where k_0^2 denotes the exact eigenvalue of the continuous problem. It is, of course, k_0^2 which is sought, however it is computationally expensive to evaluate $k^2(h)$ with small values of h. Instead, we solve the discretized problem for a series of meshes and fit a polynomial in h^2 through the mesh points. The value of the polynomial evaluated at $h = 0$ provides the Richardson extrapolation of the eigenvalue.

The extrapolation can be done with little effort. We denote the Richardson extrapolation using meshes $h_p, \ldots, h_q$ by $k^2(p, \ldots, q)$. The Richardson extrapolation for the first mesh is trivial since the polynomial fit degenerates to a constant. Thus, $k^2(1)$ is identically equal to the value obtained at the first mesh. Subsequent extrapolations are obtained recursively as follows:

$$k_j^2(p, \ldots, p+q) = \frac{(h_p^2 - h^2)k_j^2(p+1, \ldots, p+q) - (h_{p+q}^2 - h^2)k_j^2(p, \ldots, p+q-1)}{h_p^2 - h_{p+q}^2} \tag{3.39}$$

where h denotes the mesh width for which the extrapolation is desired.

As discussed in Chap. 2, a water/sediment interface can be handled by setting up an independent mesh in each medium and applying matching conditions at the interface. In such cases there are two mesh spacings h_w and h_s. As long as the ratio h_w/h_s is kept constant when the mesh is refined one can pick either $h = h_w$ or $h = h_s$ in Eq. (3.39) and obtain the same result.

The extrapolation is implemented in an adaptive fashion. An error tolerance ϵ in the eigenvalues is computed based on the maximum range for which the field will be calculated. Then, the problem is solved for successively refined meshes and at each step the extrapolation to zero mesh width ($h = 0$) is performed using the above recursion. When the difference between two successive extrapolations is less than ϵ, the extrapolated eigenvalue is accepted and the process is terminated. For simplicity, this convergence test is performed on a "key-value" which is a particular eigenvalue chosen roughly in the middle of the spectrum.

The solution of the discretized problem for each mesh can be performed as described above, however, for subsequent meshes the circumstances are changed in that we have a good estimate of the eigenvalue and eigenfunction. This information is exploited by using Richardson extrapolation in a second fashion to provide an estimate of the eigenvalue for each new mesh. In order to be confident that we extrapolations are sufficiently accurate for the root-finder this process is adopted only for the third and subsequent meshes.

A priori it is not at all obvious how well the extrapolation process should work. It depends on how rapidly the coefficients in the Taylor series for $k^2(h)$ decay. Consequently, it is necessary to examine the merits of the method for individual classes of problems. Our initial experience with smooth single layer ocean acoustic problems was extremely favorable[13]. For more complicated multilayer problems one must take care that the mesh does not straddle an interface[64]. For this reason the difference equations have been set up to allow an independent mesh within each medium.

An additional issue arises when the sound speed is provided in a tabular form as is typical for ocean acoustic problems. If piecewise-linear interpolation of the

sound speed profile is used, then the mesh must be set up to land precisely on those depths where the sound speed is tabulated. Since this must be done at each mesh, the mesh refinement is refined by simple halving. Alternatively, a new "medium" can be introduced for each piecewise linear layer. Finally, one can bypass both these alternatives by simply using a very smooth fit to the sound speed profile. In the present version, one has the option of using a spline fit, which however can introduce its own artifacts.

The reader who is having difficulty remembering the constraints and the various alternatives may be comforted to know that in practice the extrapolation works quite well even when the theoretical requirements are not satisfied. At one extreme, if you have a single medium with numerous discontinuities within the medium, you would probably find that the extrapolation provided no improvement over the answers obtained at the finest mesh. On the other hand, for extremely smooth profiles the extrapolation yields typically 3 additional significant digits per extrapolation.

We should also mention for the reader that contemplates modifying the code, that we have found that seemingly innocuous changes can introduce terms of odd power in h to the Maclaurin series for $k^2(h)$. The result then is to eliminate the usefulness of polynomial extrapolation in h^2. Such terms can easily be introduced at interfaces if care is not taken that the interface condition is $O(h^2)$ and that higher-order terms of odd power are not present.

Chapter 4

Running the Program

The KRAKEN program is actually part of a complete package of modeling tools referred to as the Acoustics Toolbox and structured as shown in Fig. 4.1. Besides the KRAKEN normal mode model, there is also a 1) ray/beam tracing model, BELLHOP, 2) an FFP (spectral integral), SCOOTER, and 3) a time-domain FFP model, SPARC.

The models take as input a user-provided environmental file (ENVFIL) to describe the problem. This file has the same format for all models. PLOTSSP can be used to produce a plot of the sound speed profile defined in the environmental file.

The models then produce a binary 'shade' file (SHDFIL) that contains calculated pressure fields. PLOTFIELD can be used to convert the pressure to transmission loss and produce a color or grey shade plot of the transmission loss over range and depth. The program PLOTSLICE is used to plot a slice of the field along a fixed receiver depth. Additional programs exist for using the shade files to do matched-field processing, to compute a probability of detection or a radius of detection.

There are utilities available for converting between the NRL shade file format and the SACLANTCEN format (TONRL and TOSAC). This allows the SACLANTCEN models to be plotted using PLOTFIELD and PLOTSLICE for intermodel comparisons. There are also utilities for converting the shade file to an ASCII format (TOASC) and back to their original binary format (TOBIN). These programs allow you to transfer files between computers with incompatible binary files by transferring an ASCII file instead.

While the basic structure is as shown in Fig. 4.1, each of the models has additional plotting routines which are unique to it. For instance, the BELLHOP ray model produces rays and so it has a ray plotting program, while the KRAKEN normal mode produces modes and so it has a mode plotting program. In this chapter, we focus on the description of the KRAKEN component, however, the other models are also discussed (briefly).

Most of the development work has been done on a VAX using VMS Fortran but careful thought has been given to portability. The following changes are necessary to run KRAKEN under UNIX using the f77 compiler:

1. Change the logical record length used for opening files. VMS uses longwords (4 bytes) most other systems seem to use bytes.

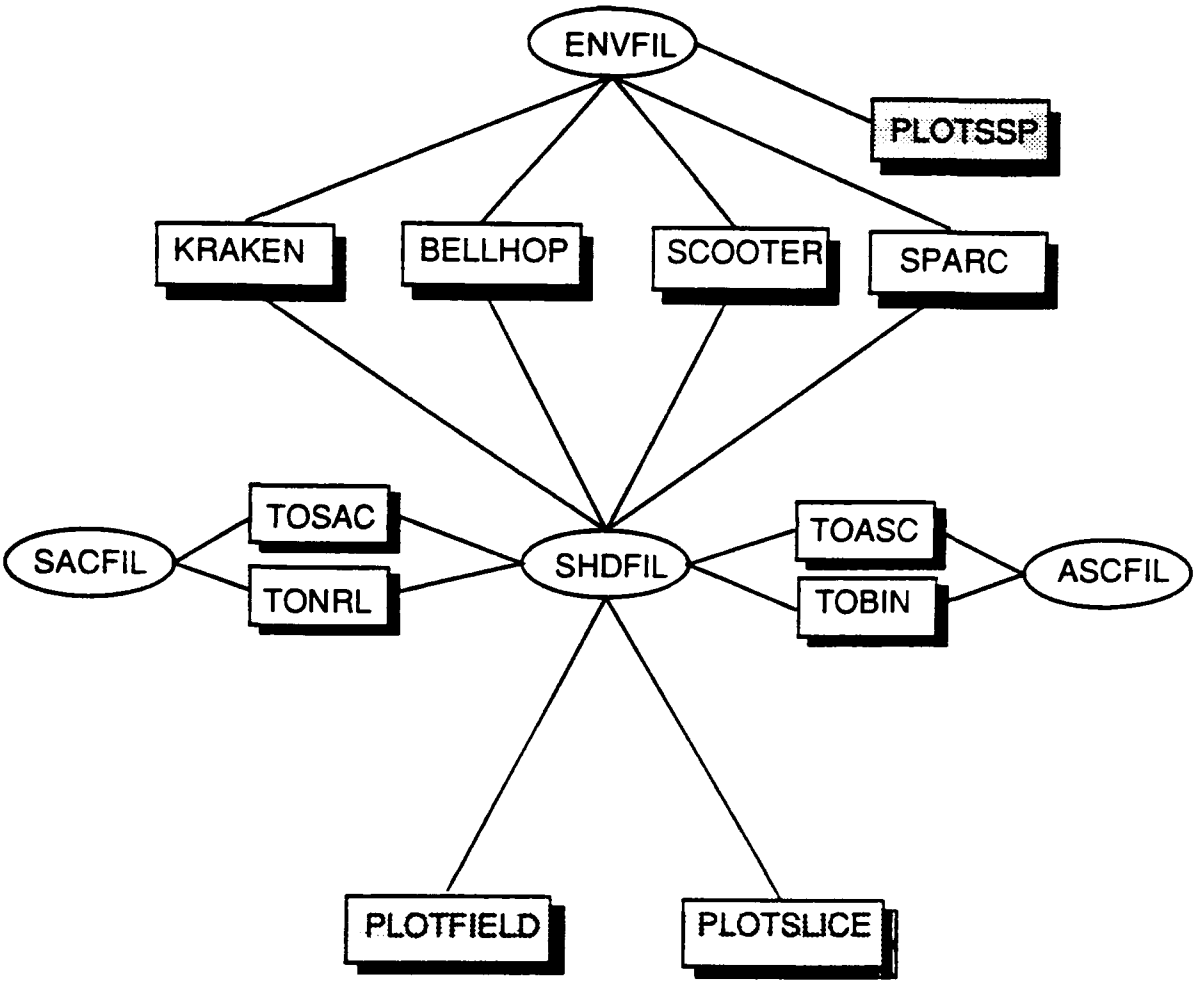


Figure 4.1: Structure of the Acoustics Toolbox.

2. Change the timing routines in TIME.FOR.

3. Change the machine constants in SLATECBESSEL.FOR *or* eliminate the Twersky ice-scatter option which uses those routines. This is done by replacing TWERSK.FOR with TWERSKYFUSE.FOR. With the latter routine it is no longer necessary to link with SLATECBESSEL and MATHIEU.

4. Apparently there is no way for the Unix system to retrieve a record length for a file automatically. You will need to modify the mode-file format to include the record length, do a preliminary read to obtain the record length, then re-open the file with the correct record length.

5. If you have core space problems, change the parameter MAXN which controls the maximum number of mesh points in depth. In KRAKEN this and other parameters are defined in the include-file COMMON.FOR . Similar include-files exist for KRAKENC, SCOOTER, SPARC, and BELLHOP.

4.1 Structure of the KRAKEN model

A schematic of the KRAKEN program structure is shown in Fig. 4.2. At the first level we see that KRAKEN actually consists of three different models KRAKEN, KRAKENC and KRAKEL. KRAKENC and KRAKEL are for more sophisticated users with special requirements. The differences are discussed in more detail in the KRAKEN.HLP file below.

A transmission loss calculation involves a two-step process running in sequence 1) KRAKEN to calculate the modes and 2) PLOTTLR or PLOTTLD to sum up the modes and plot TL versus range or depth. In addition, PLOTMODE can be run to look at the individual modes and PLOTGRN can be used to calculate a Green's function.

Producing a grey shade or color plot of transmission loss involves a three-step process running in sequence 1) KRAKEN to calculate the modes and 2) FIELD to sum the modes and calculate the pressure field, and 3) PLOTFIELD to plot the results.

Three-dimensional calculations follow a similar sequence but using FIELD3D instead of FIELD to sum the modes. As discussed in Chap. 2, 3-D calculations use a triangular patchwork over the ocean bottom that is defined in an input field-parameter file (FLPFIL). PLOTTRI is used to plot the triangular patchwork. Besides the 3-D pressure fields, FIELD3D also produces output describing the horizontal refraction which can be plotted using PLOTRAYXY.

Detailed information on how to run KRAKEN is contained in a sequence of help files included with the source code. These help files are reproduced below.

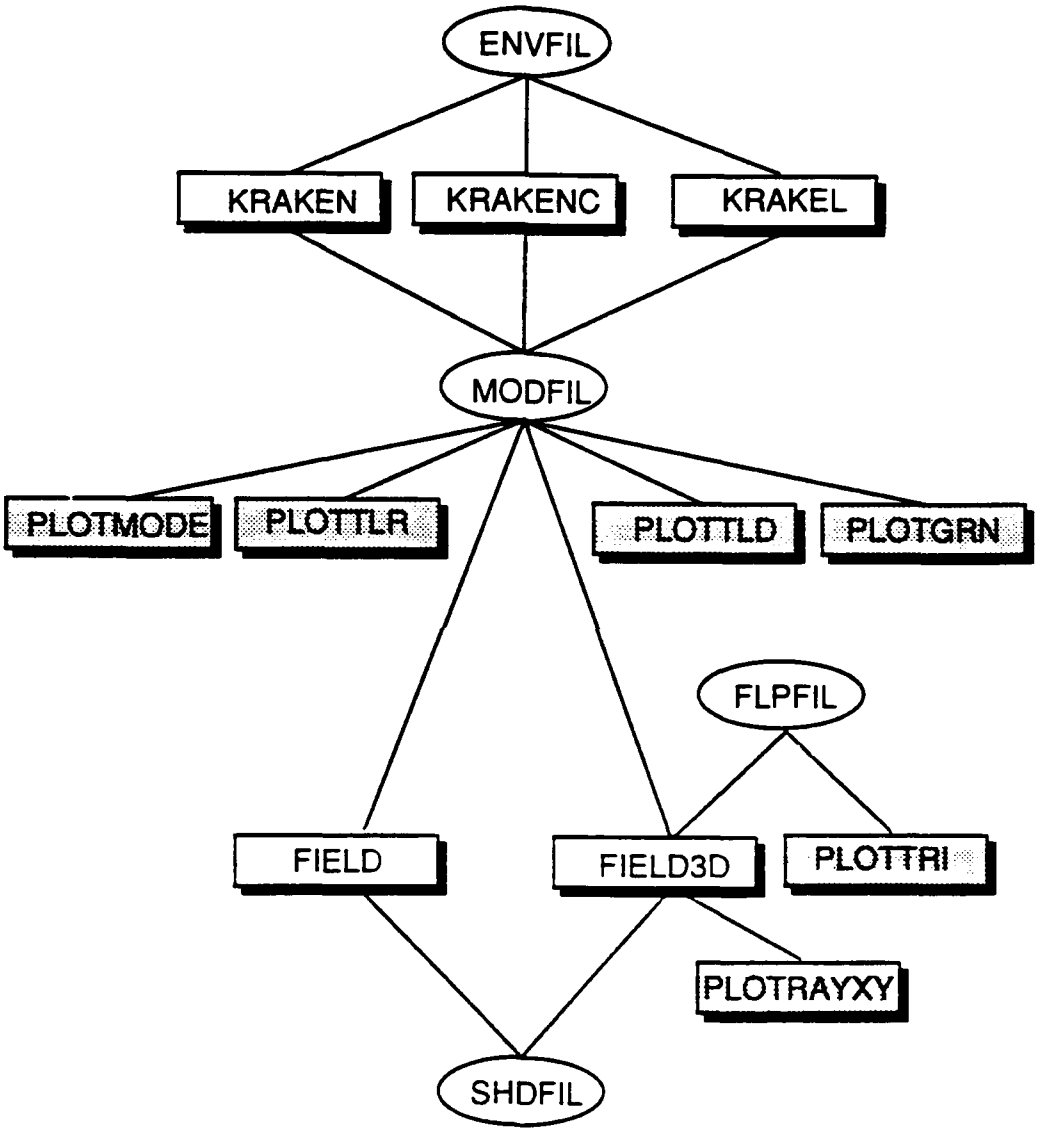


Figure 4.2: Structure of the KRAKEN model.

4.2 The Main Program

4.2.1 NOTES.HLP

KRAKEN is a normal mode program for range-varying environments in either cartesian (line sources) or cylindrical coordinates (point sources). The basic method is described in

Porter, Michael B. and Reiss, Edward L., "A numerical method for ocean-acoustic normal modes", JASA 76, 244-252 (1984).

Porter, Michael B. and Reiss, Edward L., "A numerical method for bottom interacting ocean acoustic normal modes", JASA 77, 1760-1767 (1985).

Range-dependent solutions are obtained by using optionally adiabatic or coupled mode theory.

The principal plot package used is DISSPLA. (The PLOTFIELD program actually uses the UNIRAS plot package; however, an older version, PLOTFIELDD, which uses DISSPLA, is available on disk.)

The following modules are part of the package.

```
GROUP I: MODE COMPUTATIONS:


  KRAKEN     Solves for the modes and writes them to disk.
             Elastic media are allowed but material attenuation
             in an elastic medium is ignored.

  KRAKENC    A version of KRAKEN which finds the eigenvalues in the
             complex plane.  KRAKEN uses perturbation theory to
             obtain imaginary parts of the eigenvalues while KRAKENC
             computes the complex eigenvalues exactly.

             KRAKENC runs about 3 times slower but is necessary for
             leaky mode computations or for including material
             attenuation in elastic media.  Internally KRAKENC replaces
             elastic layers by an equivalent reflection coefficent.
             For this reason, you cannot use KRAKENC to look at
             fields within the elastic layers.

  KRAKEL     Analogous to KRAKENC but also computes elastic
             displacements and stresses for elastic media.
             KRAKEL is seldom used and tends to not be kept
             up-to-date.

GROUP II: BASIC PLOTTING ROUTINES:
```

PLOTSSP Plots the sound speed profile.

PLOTMODE Plots selected modes.

PLOTGRN Plots the Green's funtion for the depth separated wave equation for a particular source/receiver combination.

PLOTTLR Plots transmission loss versus range.

PLOTTLD Plots transmission loss versus depth.

PLOTTRI Plots the triangular elements used for 3-D field calculations.

GROUP III: FIELD COMPUTATIONS:

FIELD Computes fields on a vertical array over a specified range and for a series of source depths. Individual phones in the array may be displaced from the vertical. Range dependence is handled by either adiabatic or one-way coupled mode theory.

FIELD3D Computes field for a three-dimensionally varying SSP using adiabatic mode theory.

GROUP IV: PLOTTING ROUTINES THAT USE GROUP III PROGRAM OUTPUT:

PLOTFIELD Plots tranmission loss in plan or elevation, i.e. an (x,y) plot or an (r,z) plot.

PLOTSLICE Plots overlays of transmission loss versus range curves by extracting slices from several shade files.

PLOTRAYXY Plots the ray paths of the Gaussian beams generated during 3D field calculations.

The various programs for computing fields (GROUP III) are only needed for PLOTFIELD, or for special user programs (e.g. ambiguity surfaces). PLOTTLR and PLOTTLD compute the field internally and therefore do not need a shade file from FIELD to

run.

The following extensions are used with these programs:

.FOR	The FORtran source code
.HLP	A HeLP file documenting the module
.COM	A COMmand file which runs the module
.LNK	A command file which performs a LiNK

All user input in all modules is read using list-directed I/O. Thus data can be typed in free-format using space, tabs, commas or slashes as delimeters. Character input should be enclosed in single quotes like this: 'CHARACTER INPUT'.

You will see the '/' character in a number of the input files. This terminates an input line causing the program to use default values.

***** INSTALLATION NOTES *****

There is a VMS command file for each of the programs in this package which assigns necessary input files to the appropriate Fortran unit number used by that program. In order to simplify the installation, these command files make use of logical names for certain directories. The logical names are in turn defined in a single file call AT_INIT.COM which is the ONLY file which needs to be customized for a new VMS installation.

The following symbols and logical names for directories are used with the KRAKEN command files:

AT: This is the Acoustics Toolbox directory which contains command files for running KRAKEN and other models in the toolbox.

KRAK: The KRAKEN source code

MISC: Miscellaneous scientific subroutines, e.g. root-finders, linear equation solvers, ...

GLOB: Global routines, that is, routines which operate on shade files. These routines operate on the output of a number of different propagation codes including

KRAKEN, FSTFLD, BELLHOP, SCOOTER and SPARC.

SCR: A directory for scratch files.

DISSPLA is a symbol which points to the DISSPLA plotting library.

The following is an example of how these might be defined under:

```
$ DEFINE    AT          US:[PORTER.AT]
$  DEFINE   BELL        US:[PORTER.AT.BELLHOP]
$  DEFINE   GLOB        US:[PORTER.AT.GLOBAL]
$  DEFINE   KRAK        US:[PORTER.AT.KRAKEN]
$  DEFINE   MAN         US:[PORTER.AT.MANUAL]
$  DEFINE   MISC        US:[PORTER.AT.MISC]
$  DEFINE   SCO         US:[PORTER.AT.SCOOTER]
$ DEFINE  SCR           US:[PORTER.SCR]
$ !
$ DISSPLA == "[DIS11.LIB]DISLIB/L, INTLIB/L, DISLIB/L, HCBS/L"
```

***** HOW TO RUN KRAKEN *****

0. Starting out for the first time? Take a look at CLINK.COM for a compile and link of the whole package.

1. Create the environmental file for your problem, following the directions in KRAKEN.HLP.

2. Run KRAKEN (or KRAKENC). On the VAX this is done by typing either

   ```
   @KRAKEN filename
      or
   SUBMIT KRAKEN/PAR=filename
   ```

   where "filename" is the environmental file.
   The KRAKEN.HLP file details the differences between the KRAKEN and KRAKENC.

3. You now have several choices (all the GROUP II programs):

a. Plot tranmission loss:

```
@PLOTTLR filename
```

b. Plot the modes:

```
@PLOTMODE filename
```

c. Plot the sound speed profile (actually, this can be done even before running KRAKEN):

```
@PLOTSSP filename
```

d. Plot the pressure field as a function of range and depth. This is a 2-step process:

```
@FIELD filename
@PLOTFIELD filename
```

In general, you'll have to modify each command file before running it to provide the appropriate inputs as described in the help file for each program.

Once the modes are created by KRAKEN or KRAKENC you can run the above plot programs in any sequence or as often as you like.

4.2.2 KRAKEN.HLP

KRAKEN is the main program. It takes an environmental file, computes the modes, and writes them to disk for use by other modules. A print file is also produced, echoing the user input.

KRAKENC is a complex arithmetic version (hence the C in KRAKENC) of KRAKEN. By working in the complex domain, loss mechanisms such as ice scatter and material absorption may be included 'exactly' rather than perturbatively. In addition, leaky modes may be computed. The price of this non-perturbative treatment is a slowdown in speed by approximately a factor of 4. This factor principally represents the difference between complex and real arithmetic.

A further slow down by a factor of 2 or more may occur it the Twersky scatter option is used in KRAKENC. The calculation of the Twersky scatter function can require significant CPU time; enough to actually be a dominant part of the cost of computing the modes. KRAKEN incorporates the scatter perturbatively and is much less sensitive to the cost of Twersky scatter.

KRAKEN does not at allow for losses in elastic media due to material attenuation. Thus, for attenuating elastic media, KRAKENC should be used.

```
Files:

        Name            Unit        Description
Input
        *.ENV            1       ENVironmental data
        *.BRC           10       Bottom   Refl. Coef.  (optl)
        *.TRC           11       Top      Refl. Coef.  (optl)
        *.IRC           12       Internal Refl. Coef.  (optl)
Output
        *.PRT            6       PRinT file
        *.MOD           20       MODe  file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF ENV FILE:

 'FRAMIV Twersky S/S ice scatter'          ! TITLE
 50.0                                      ! FREQ (Hz)
 4                                ! NMEDIA
 'NSF'                            ! OPTIONS
 0.0092  8.2  5.1                 ! BUMDEN (1/m)  ETA (m)  XI (m)
 750  0.0  3750.0                 ! NMESH  SIGMA (m)  NSSP
     0.0  1436.0  0.0  1.03/      ! Z(m)  CP  CS(m/s)  RHO(gm/cm3)
    30.0  1437.4 /
    50.0  1437.7 /
    80.0  1439.5 /
```

```
   100.0  1441.9 /
   125.0  1444.6 /
   150.0  1450.0 /
   175.0  1456.1 /
   200.0  1458.4 /
   250.0  1460.0 /
   300.0  1460.5 /
   350.0  1460.6 /
   400.0  1461.0 /
   450.0  1461.5 /
   500.0  1462.0 /
   600.0  1462.9 /
   700.0  1463.9 /
   800.0  1464.8 /
   900.0  1465.8 /
  1000.0  1466.7 /
  1100.0  1467.0 /
  1200.0  1469.0 /
  1300.0  1469.5 /
  1400.0  1471.8 /
  1600.0  1474.5 /
  1800.0  1477.0 /
  2000.0  1479.6 /
  2500.0  1487.9 /
  3750.0  1510.4 /
 35  0.0  3808.33
  3750.0  1504.6     0.0   1.50   .15  0.0
  3808.33 1603.07 /
 35  0.0  3866.66
  3808.33 1603.07    0.0   1.533  .15  0.0
  3866.66 1701.53 /
 35  0.0  3925.0
  3866.66 1701.53    0.0   1.566  .15  0.0
  3925.0  1800.0 /
 'A'  0.0                        ! BOTOPT  SIGMA (m)
  3925.0  1800.0     0.0   1.60   .15  0.0
 0.0  1504.0                     ! CLOW  CHIGH (m/s)
 300.0                           ! RMAX (km)
 1 100.0 /                       ! NSD  SD(1:NSD) (m)
 1 200.0 /                       ! NRD  RD(1:NRD) (m)
```

DESCRIPTION OF INPUTS:

(1) - TITLE

Syntax:
TITLE
Description:
TITLE: Title of run enclosed in sinqle quotes.

(2) - FREQUENCY

Syntax:
FREQ
Description:
FREQ: Frequency in Hz.

(3) - NUMBER OF MEDIA

Syntax:
NMEDIA (<20)
Description:
NMEDIA: Number of media.

The problem is divided into media within which it is assumed that the material properties vary smoothly. A new medium should be used at fluid/elastic interfaces or at interfaces where the density changes discontinuously. The number of media in the problem is defined excluding the upper and lower half-space.

(4) - OPTIONS

Syntax:
OPTION
Description:
OPT(1:1): Type of interpolation to be used for the SSP.
'C' for C-linear,
'N' for N2-linear (n the index of refraction),
'S' for cubic Spline,
'A' for Analytic. The user must modify the analytic formulas in PROFIL.FOR then compile and link.

If your not sure which option to take, I'd suggest you use 'C' or 'N'. Practically, you can pick either one: the choice has been implemented to facilitate precise intermodel comparisons.

Option 'S' is a little dangerous because splines yield a poor fit to certain kinds of curves, e.g. curves with sharp bends. If you insist on splines, you can fix a bad fit by dividing the water column into two 'media' at the bend.

Run PLOTSSP to check that the SSP looks the way you thought it should. Apart from potential typos, this will also show up fit-problems.

OPT(2:2): Type of top boundary condition.

- 'V' VACUUM above top.
- 'A' ACOUSTO-ELASTIC half-space. Requires another line as described in block (4a).
- 'R' Perfectly RIGID.
- 'F' Reflection coefficient from a FILE (available in KRAKENC only). Requires additional lines as described in block (4c).
- 'W' WRITE an internal reflection coefficient to a file (available in KRAKENC only). The file is given the extension '.IRC' and can subsequently be read in as a bottom boundary condition. (See option 'P' under bottom boundary conditons.)
- 'S' for Soft-boss Twersky scatter.
- 'H' for Hard-boss Twersky scatter.
- 'T' for Soft-boss Twersky scatter, amplitude only.
- 'I' for Hard-boss Twersky scatter, amplitude only. The Twersky scatter options require another line as described in block (4c). Mnemonically, T, I options are one letter after S, H in the alphabet. Current wisdom is that option T is most appropriate for ice scatter.

For open ocean problems option 'V' should be

```
                used for the top BC.  The Twersky options
                are intended for under-ice modeling.

        OPT(3:3): Attenuation units.
                  'N' Nepers/m.
                  'F' dB/(kmHz)        (F as in Freq. dependent)
                  'M' dB/m             (M as in per Meter)
                  'W' dB/wavelength    (W as in per Wavelength)
                  'Q' quality factor.
                  'T' Thorp attenuation formula.  This overrides
                      any other attenuations specified.

                  KRAKEN ignores material attenuation
                  in elastic media. (KRAKENC treats
                  it properly).

        OPT(4:4): Slow/robust root-finder.
                  '.' As in: I want all the modes and I don't
                      care how long it takes. Period.
                      (Available in KRAKENC only.)
                      In certain problems with elastic layers
                      the old root-finder has been known to
                      skip modes.


(4a) - TOP HALFSPACE PROPERTIES

      Syntax:
         ZT  CPT  CST  RHOT  APT  AST
      Description:
         ZT:   Depth (m).
         CPT:  Top P-wave speed (m/s).
         CST:  Top S-wave speed (m/s).
         RHOT: Top density (g/cm3).
         APT:  Top P-wave attenuation. (units as given in Block 2)
         AST:  Top S-wave attenuation. (  "   "    "    "   "   ")

         This line should only be included if OPT(2:2)='A', i.e.
         if the user has specified a homogeneous halfspace for
         the top BC.


(4b) - TOP REFLECTION COEFFICIENT

      Syntax:
```

```
   NTHETA
   THETA(1)        RMAG(1)        RPHASE(1)
   THETA(2)        RMAG(2)        RPHASE(2)
     .
     .
     .
   THETA(NTHETA)  RMAG(NTHETA)  RPHASE(NTHETA)
```

Description:

NTHETA: Number of angles.
THETA(): Angle.
RMAG(): Magnitude of reflection coefficient.
RPHASE(): Phase of reflection coefficient (degrees).

Example:

```
3
0.0   1.00  180.0
45.0  0.95  175.0
90.0  0.90  170.0
```

These lines should be contained in a separate '.TRC' file. This file is only required if OPT(2:2)='F', i.e. if the user has specified that the top BC is read from a '.TRC' (Top Reflection Coefficient) file.

This option for tabulated reflection coefficients is somewhat experimental at this time. I haven't worried about the multivalued character of the phase function: choose your reference and make sure the phase varies continuously. A complicated reflection coefficient may well cause problems for the mode-finder.

(4c) - TWERSKY SCATTER PARAMETERS

Syntax:

```
   BUMDEN  ETA  XI
```

Description:

BUMDEN: Bump density (ridges/km).
ETA: Principal radius 1 (m).
XI: Principal radius 2 (m).

This line should only be included when one of the Twersky-scatter options is selected.

(5) - MEDIUM INFO

Syntax:

```
NMESH  SIGMA  Z(NSSP)
```

Description:

NMESH: Number of mesh points to use initially. The number of mesh points should be about 10 per vertical wavelength in acoustic media. In elastic media, the number needed can vary quite a bit; 20 per wavelength is a reasonable starting point.

The maximum allowable number of mesh points is given by 'MAXN' in the dimension statements. At present 'MAXN' is 50000. The number of mesh points used depends on the initial mesh and the number of times it is refined (doubled). The number of mesh doublings can vary from 1 to 5 depending on the parameter RMAX described below.

SIGMA: RMS roughness at the interface.

Z(NSSP): Depth at bottom of medium (m). This value is used to detect the last SSP point when reading in the profile that follows.

(5a) - SOUND SPEED PROFILE

Syntax:

```
Z(1)     CP(1)     CS(1)     RHO(1)     AP(1)     AS(1)
Z(2)     CP(2)     CS(2)     RHO(2)     AP(2)     AS(2)
  .
  .
  .
Z(NSSP)  CP(NSSP)  CS(NSSP)  RHO(NSSP)  AP(NSSP)  AS(NSSP)
```

Description:

Z(): Depth (m). The surface starts at the first depth point specified. Thus if you have say, XBT data which starts at 50 m below the surface, then you'll need to put in some SSP point at 0 m, otherwise the free-surface would be placed at 50 m giving erroneous results.

```
CP():    P-wave speed (m/s).
CS():    S-wave speed (m/s).
RHO():   Density (g/cm3).
         Density variations within an acoustic medium
         are at present ignored.
AP():    P-wave attenuation (units as given in Block 2)
AS():    S-wave attenuation (  "   "    "    "   "   ")
```

These lines should be omitted when the 'A' option is used (indicating that an analytic profile is supplied by a user written subroutine).

The '/' character signals that the remaining data on the line is the same as in the previous line of SSP data. For the very first line the default or 'previous' line is:

```
0.0 1500.0 0.0 1.0 0.0 0.0
```

This block should be repeated for each subsequent medium.

(6) - BOTTOM BOUNDARY CONDITION

```
Syntax:
   BOTOPT  SIGMA
Description:
   BOTOPT: Type of bottom boundary condition.
           'V' VACUUM below bottom.
           'A' ACOUSTO-ELASTIC half-space.
               Requires another line with the half-space
               parameters.  The format is the same as that
               used for specifying the top halfspace BC.
           'R' Perfectly RIGID.
           'F' reflection coefficient from a FILE (available
               in KRAKENC only). Requires a Bottom
               Reflection Coefficient file with
               extension '.BRC'.  The format is the same as
               that used for a Top Reflection coefficient.
           'P' Precaculated internal reflection coefficient
               from a FILE (available in KRAKENC only)
           Option 'A' is generally used for ocean bottom
           modeling.
   SIGMA:  Interfacial roughness (m).
```

(7) - PHASE SPEED LIMITS

Syntax:

CLOW CHIGH

Description:

CLOW: Lower phase speed limit (m/s).
CLOW will be computed automatically if you set it to zero. However, by using a nonzero CLOW you can skip the computation of slower modes. Mainly this is used to exclude interfacial modes (e.g. a Scholte wave). The root finder is especially slow in converging to these interfacial modes and when the source and receiver are sufficiently are far from the interface the interfacial modes are negligible.

CHIGH: Upper phase speed limit (m/s).
The larger CHIGH is, the more modes are calculated and the longer the execution time. Therefore CHIGH should be set as small as possible to minimize execution time.

On the other hand, CHIGH controls the maximum ray angle included in a subsequent field calculation-- ray paths are included which turn at the depth corresponding to CHIGH in the SSP. Thus a larger CHIGH means more deeply penetrating rays are included.

Choice of CHIGH then becomes a matter of experience. In the far-field and at high-frequencies, rays travelling in the ocean bottom are severely attenuated and one may set CHIGH to the sound speed at the ocean bottom. In the near-field, low-frequency case, rays refracted in the bottom may contribute significantly to the field and CHIGH should be chosen to include such ray paths.

KRAKEN will (if necessary) reduce CHIGH so that only trapped (non-leaky) modes are computed.

KRAKENC will attempt to compute leaky modes if CHIGH exceeds the phase velocity of either the

S-wave or P-wave speed in the half-space. Leaky mode computations are somewhat experimental at this time.

(8) - MAXIMUM RANGE

Syntax:

RMAX

Description:

RMAX: Maximum range (km).
This parameter should be set to the largest range for which a field calculation will be desired.

During the mode calculation the mesh is doubled successively until the eigenvalues are sufficiently accurate at this range. If you set it to zero, then no mesh doublings will be performed. You don't need to worry too much about this parameter-- even if you set it to zero the results will usually be reasonable.

(9) - SOURCE/RECEIVER DEPTH INFO

Syntax:

NSD SD(1:NSD)
NRD RD(1:NRD)

Description:

NSD: The number of source depths.
SD(): The source depths (m).
NRD: The number of receiver depths.
RD(): The receiver depths (m).

This data is read in using list-directed I/O so you can type it just about any way you want, e.g. on one line or split onto several lines. Also if your depths are equally spaced then you can type just the first and last depths followed by a '/' and the intermediate depths will be generated automatically.

CPU time is essentially independent of the number of sources and receivers so that you can freely ask for up to 4095 depths. However, for high-frequencies the storage for the mode files can be excessive.

The source/rcvr depths are sorted and merged and then the modes are calculated at the union of the two sets of depths. Thus, it doesn't matter if you mix up source and receiver depths. Furthermore, you can leave out either the source or receiver specification (but not both simultaneously) simply by using a '/' for that line.

Sources and receivers cannot be placed in a half-space.

SAMPLE PRINT OUT

The print-out for this deck is shown below

```
KRAKEN- FRAMIV Twersky S/S ice scatter
Frequency =   20.00     NMEDIA =   4


    N2-LINEAR approximation to SSP
    Attenuation units: dB/mkHz
    TWERSKY SOFT BOSS surface scatter model

Twersky ice model parameters:
Bumden =    0.920000E-02 Eta =    8.20     Xi =    5.10


     Z          ALPHAR      BETAR       RHO        ALPHAI      BETAI


          ( Number of pts =   750  RMS roughness =  0.000E+00 )
     0.00       1436.00       0.00      1.03        0.0000     0.0000
    30.00       1437.40       0.00      1.03        0.0000     0.0000
    50.00       1437.70       0.00      1.03        0.0000     0.0000
    80.00       1439.50       0.00      1.03        0.0000     0.0000
   100.00       1441.90       0.00      1.03        0.0000     0.0000
   125.00       1444.60       0.00      1.03        0.0000     0.0000
   150.00       1450.00       0.00      1.03        0.0000     0.0000
   175.00       1456.10       0.00      1.03        0.0000     0.0000
   200.00       1458.40       0.00      1.03        0.0000     0.0000
   250.00       1460.00       0.00      1.03        0.0000     0.0000
```

```
   300.00      1460.50      0.00      1.03      0.0000     0.0000
   350.00      1460.60      0.00      1.03      0.0000     0.0000
   400.00      1461.00      0.00      1.03      0.0000     0.0000
   450.00      1461.50      0.00      1.03      0.0000     0.0000
   500.00      1462.00      0.00      1.03      0.0000     0.0000
   600.00      1462.90      0.00      1.03      0.0000     0.0000
   700.00      1463.90      0.00      1.03      0.0000     0.0000
   800.00      1464.80      0.00      1.03      0.0000     0.0000
   900.00      1465.80      0.00      1.03      0.0000     0.0000
  1000.00      1466.70      0.00      1.03      0.0000     0.0000
  1100.00      1467.00      0.00      1.03      0.0000     0.0000
  1200.00      1469.00      0.00      1.03      0.0000     0.0000
  1300.00      1469.50      0.00      1.03      0.0000     0.0000
  1400.00      1471.80      0.00      1.03      0.0000     0.0000
  1600.00      1474.50      0.00      1.03      0.0000     0.0000
  1800.00      1477.00      0.00      1.03      0.0000     0.0000
  2000.00      1479.60      0.00      1.03      0.0000     0.0000
  2500.00      1487.90      0.00      1.03      0.0000     0.0000
  3750.00      1510.40      0.00      1.03      0.0000     0.0000

        ( Number of pts =    35  RMS roughness =  0.000E+00 )
  3750.00      1504.60      0.00      1.50      0.1500     0.0000
  3808.33      1603.07      0.00      1.50      0.1500     0.0000

        ( Number of pts =    35  RMS roughness =  0.000E+00 )
  3808.33      1603.07      0.00      1.53      0.1500     0.0000
  3866.66      1701.53      0.00      1.53      0.1500     0.0000

        ( Number of pts =    35  RMS roughness =  0.000E+00 )
  3866.66      1701.53      0.00      1.57      0.1500     0.0000
  3925.00      1800.00      0.00      1.57      0.1500     0.0000

   ACOUSTO-ELASTIC half-space,  ( RMS roughness =  0.000E+00 )
  3925.00      1800.00      0.00      1.60      0.1500     0.0000

CLOW =  0.00000E+00 CHIGH =   1504.0
RMAX =    300.0000000000000

Number of sources   =           1
  100.0000

Number of receivers =           1
  200.0000

Mesh multiplier   CPU seconds
```

```
        1              16.4
        2              15.1

  I          K                    ALPHA          PHASE SPEED
  1   0.8625082052E-01  -0.8519020992E-06    1456.956646
  2   0.8582849772E-01  -0.1302695655E-06    1464.125663
  3   0.8562855085E-01  -0.1059327457E-06    1467.544468
  4   0.8545402623E-01  -0.1136748056E-06    1470.541667
  5   0.8527187871E-01  -0.1192384459E-06    1473.682861
  6   0.8510445198E-01  -0.1156165482E-06    1476.582050
  7   0.8495255965E-01  -0.1130917467E-06    1479.222129
  8   0.8479984039E-01  -0.1185453302E-06    1481.886116
  9   0.8465149335E-01  -0.1314814525E-06    1484.483039
 10   0.8450452348E-01  -0.1255743704E-06    1487.064845
 11   0.8435857532E-01  -0.1276318031E-06    1489.637606
 12   0.8421637950E-01  -0.1377681231E-06    1492.152796
 13   0.8407780307E-01  -0.1377169389E-06    1494.612151
 14   0.8393959060E-01  -0.1339925824E-06    1497.073136
 15   0.8380370528E-01  -0.1378254389E-06    1499.500598
 16   0.8367091002E-01  -0.1450063419E-06    1501.880476
```

If the program aborts in some way, examine the print file which is produced. Frequently an expected line has been omitted and the environmental file is therefore misinterpreted.

The message "FAILURE TO CONVERGE IN SECANT" occurs when KRAKEN requires more than 500 iterations to converge to a mode. Usually less than 20 iterations are needed but convergence to interfacial modes (Scholte or Stoneley waves) can be exceptionally slow, especially at higher frequencies. The simplest solution is to exclude interfacial modes by setting the lower phase-speed limit to the minimum p-wave speed in the problem. Alternately, you can increase the value of MAXNIT which controls the MAXimum Number of ITerations in the root finder.

4.3 Acoustic Field Calculations

4.3.1 FIELD.HLP

The FIELD program uses the modes calculated by KRAKEN and produces a shade file which contains a sequence of snapshots of the acoustic field as a function of range and depth. A snapshot is produced for every source depth specified by the user.

```
Files:

        Name            Unit         Description
Input
        *.FLP             5        FieLd Parameters
        *.MOD           30-99      MODe files

Output
        *.PRT             6        PRinT file
        *.SHD            25        SHaDe file

------------------------------------------------------------

EXAMPLE AND DESCRIPTION OF FLP FILE:

/,                                  ! TITLE
'RA'                                ! OPT 'X/R', 'C/A'
9999                                ! M  (number of modes to include)
1     0.0                           ! NPROF   RPROF(1:NPROF) (km)
501  200.0  220.0 /                 ! NR    R(1:NR)   (km)
1   500.0 /                         ! NSD  SD(1:NSD)   (m)
1  2500.0 /                         ! NRD  RD(1:NRD)   (m)
1     0.0 /                         ! NRR  RR(1:NRR)   (m)


 (1) - TITLE

       Syntax:
          TITLE
       Description:
          TITLE: Title to be written to the shade file.
                 If you type a /, the title is taken from the
                 first mode file.

 (2) - OPTIONS

       Syntax:
```

```
   OPTION
Description:
   OPTION(1:1): Source type.
                'R' point source
                    (cylindrical (R-Z) coordinates)
                'X' line  source
                    (cartesian   (X-Z) coordinates)
   OPTION(2:2): Selects coupled or adiabatic mode theory.
                'C' Coupled mode theory.
                'A' Adiabatic mode theory (default).
```

(3) - NUMBER OF MODES

```
Syntax:
   M
Description:
   M:    Number of modes to use in the field computation.
         If the number of modes specified exceeds the
         number computed then the program uses all the
         computed modes.
```

(4) - PROFILE RANGES

```
Syntax:
   NPROF  RPROF(1:NPROF)
Description:
   NPROF:   The number of profiles, i.e. ranges where a new
            set of modes is to be used.
   RPROF(): Ranges (km) of each of these profiles.
            For a range independent problem there is only
            one profile and its range is arbitrary.
            mode files must exist for each range of a
            new profile and be assigned in sequence to
            units 30,31,... The modes for the last SSP
            profile are extended in a range-independent
            fashion to infinity so that RMAX can exceed
            RPROF(NPROF).
```

(6) - SOURCE/RECEIVER LOCATIONS

```
Syntax:
   NR    R(1:NR)
```

```
   NSD  SD(1:NSD)
   NRD  RD(1:NRD)
   NRR  RR(1:NRR)
Description:
   NR:   Number of receiver ranges.
         (NR<4094 and NR*NRD <= 210000)
   R():  The receiver ranges (km)
   NSD:  The number of source depths. (<51)
   SD(): The source depths (m).
   NRD:  The number of receiver depths.
         (<201 and NR*NRD < 210000)
   RD(): The receiver depths (m).
   NRR:  The number of receiver range-displacements.
         Must equal NRD. (YES, IT IS REDUNDANT)
   RR(): The receiver displacements (m).
         This vector should be all zeros for a perfectly
         vertical array.

         The  field is computed by stepping through the
         ranges, R(1:NR), and adding in the range
         displacements, RR() before computing the field
         on the array. Nonzero values are used to tilt or
         distort the receiving array, thereby simulating the
         distortion which occurs on an array deployed in
         the ocean.


The format of the source/rcvr info is an integer
indicating the number of sources (receivers) followed by
real numbers indicating the depth (range) of each
receiver.  Since this data is read in using list-directed
I/O you can type it just about any way you want, e.g. on
one line or split onto several lines.  Also if your depths
are equally spaced then you can type just the first and
last depths followed by a '/' and the intermediate depths
will be generated automatically.
```

4.3.2 FIELD3D.HLP

The FIELD3D program uses the modes calculated by KRAKEN and produces a shade file which contains a sequence of 2-D slices of the acoustic field. It is commonly used to compute a field in plan view, i.e. as a function of horizontal coordinates (x,y). It can also be used to compute the field on a vertical slice along any fixed bearing through the 3-D environment.

FIELD3D uses a tiling of the ocean environment based on triangles. The terminology is taken from finite-elements. To define the triangles you must do the following:

(1) Lay out a grid of points (nodes) where you will construct environmental files for KRAKEN and solve for the modes. A coarse rule-of-thumb is to pick points every 10 km but obviously a coarser spacing can be used in sites with less environmental change.

(2) Assign a number to each of the nodes.

(3) Form a triangulation of the nodes. That is, connect the nodes with lines such that the grid is divided into a number of triangles. This should be done with an eye towards keeping the area of the individual triangles uniform. All nodes should be a corner of at least one triangle. Each triangle is referred to as an element.

There are algorithms for performing this step automatically and if you write one I would be glad to receive it. If instead you do this by hand you will rapidly discover the merits of using a regular grid.

(4) Assign a number to each of the elements.

You now have the information required by FIELD3D to describe your triangulation. In the input file you first tell FIELD3D the coordinates of each node and the name of the file containing the modes at each node. You then tell FIELD3D how you connected the nodes to form a triangulation. This is done by specifying the node numbers which define the corners of each successive element (triangle).

```
Files:

        Name            Unit          Description
Input
        *.FLP             5        FieLd Parameters
        *.MOD           30-99      MODe files

Output
        *.PRT             6        PRinT file
        *.SHD            25        SHaDe file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF FLP FILE:

'MUNK3D'                          ! TITLE
```

```
'STDFM'                              ! OPT
9999                                 ! M (number of modes)
0.001  0.001                         ! XS  YS (source position) (km)
1  1000.0                            ! NSD  SD(1:NSD) (m)
1   800.0                            ! NRD  RD(1:NRD) (m)
0.0  100.0  501                      ! RMIN  RMAX  NR (km)
19  0.0  360.0 /                     ! NTHETA  THETA(1:NTHETA) (degrees)
5                                    ! Number of SSP's (NSSP)
 100.0     0.0  'SCR:MUNKT0'         ! (x, y) i=1, NSSP (km)
   0.0   100.0  'SCR:MUNKT90'
-100.0     0.0  'SCR:MUNKT0'
   0.0  -100.0  'SCR:MUNKT270'
   0.0     0.0  'SCR:MUNKT0'
4                                    ! NELTS
5  1  2                              ! Nodes of corners
5  2  3
5  3  4
5  4  1
4.0  360.0  90                       ! ALPHA1  ALPHA2  NALPHA
500.0  160                           ! STEP  NSTEPS
0.3                                  ! EPMULT
```

(1) - OPTIONS

Syntax:
OPT
Description:
OPT(3:3): Type of caculation.
'STD' (Standard) for an Nx2D run.
'GBT' (Gaussian beam trace) for a 3D run.
'PDQ' For a fast preview run.
The 'STD' option neglects horizontal
refraction but runs a lot faster.
Avoid using the 'GBT' option:
it requires some care to use
properly. Option 'PDQ' runs about 3x as fast
as 'STD' but is less accurate.
OPT(4:4): TESCHECK (tesselation check) flag.
'T' Perform the tesselation check.
'F' omit the tesselation check.
For all but the simplest setups the user will
INVARIABLY make an error in setting up the
triangulation. The first step to avoid this
is to run PLOTTRI to get a plot of the

triangulation. Even after that one should invoke this 'TESCHECK' option however for large problems some time can be saved by turning off this feature after the triangulation has been checked once.

OPT(5:5): Type of beams.
There are several types of Gaussian beams available. I suggest using 'M'. This option is ignored unless the Gaussian beam calculation has been selected.

OPT(6:6): Ray file flag.
Use 'R' to have a file of ray path trajectories (in the horizontal plane) written to disk for subsequent plotting using the PLOTRAYXY program. These rays show the horizontal refraction of individual modes. This option is ignored if you select a 'STD' or 'PDQ' run for then the ray paths are just straight lines.

(2) - NUMBER OF MODES

Syntax:
M
Description:
M: Number of modes to use in the field computation. If the number of modes specified exceeds the number computed then the program uses all the computed modes.

(3) - SOURCE COORDINATES

Syntax:
XS YS
Description:
XS: X-coordinate of source (km).
YS: Y-coordinate of source (km).

(4) - SOURCE/RECEIVER DEPTHS

Syntax:
NSD SD(1:NSD)

```
   NRD  RD(1:NRD)
Description:
   NSD:  The number of source depths. (<3)
   SD(): The source   depths (m).
   NRD:  The number of receiver depths.
         (<51 and NR*NRD < 54000)
         Multiple receiver depths and multiple
         azimuthal radials are exclusive.  If the number of
         receiver depths is greater than one then
         the program will overide your specification of
         multiple radials.
   RD(): The receiver depths (m).
```

(5) - RECEIVER RANGES

```
Syntax:
   RMIN  RMAX  NR
Description:
   RMIN: First receiver range (km). MUST BE ZERO!
   RMAX: Last  receiver range (km).
   NR:   Number of receiver ranges.
         (NR<4094 and NR*NRD <= 210000)
```

(6) - RADIALS

```
Syntax:
   NTHETA  THETA(1:NTHETA)
Description:
   NTHETA:  Number of radials. (<101)
   THETA(): Angles for each radial (degrees).
```

For full circle (or disc) coverage our plotting program likes to have a repeated radial, say 0 and 360 degrees.

(7) - NODES

```
Syntax:
   NNODES
   X(1)       Y(1)       FILNAM(1)
   X(2)       Y(2)       FILNAM(2)
    .
    .
```

```
      .
   X(NNODES)  Y(NNODES)  FILNAM(NNODES)
Description:
   NNODES:    Number of nodes. (<1000).
   X():       X-coordinate of node (km).
   Y():       Y-coordinate of node (km).
   FILNAM():  Name of the mode file for that node.
              Use the name 'DUMMY' to produce an acoustic
              absorber.
```

(8) - ELEMENTS

```
Syntax:
   NELTS
   NODE1(1)       NODE2(1)       NODE3(1)
   NODE1(2)       NODE2(2)       NODE3(2)
     .
     .
     .
   NODE1(NELTS)  NODE2(NELTS)  NODE3(NELTS)
Description:
   NELTS:    Number of elements. (<1500)
   NODE1():  Number of node at first corner of the triangle.
   NODE2():     "   "   "    " second   "   "    "    "
   NODE3():     "   "   "    " third    "   "    "    "

   In this fashion we define a tiling of triangular
   elements. The ordering of the elements is arbitrary.
```

(9) - GAUSSIAN BEAM INFO

```
Syntax:
   ALPHA1  ALPHA2  NALPHA
   STEP  NSTEPS
   EPMULT
Description:
   ALPHA1: First angle for beam fan (degrees).
   ALPHA2: Last    "    "    "   "      "
   NALPHA: Number of beams in fan.
   STEP:   Step size (m).
   NSTEPS: Number of steps.
   EPMULT: Epsilon multipler for beam initial conditions.

This Gaussian beam info can be omitted if the 'STD' option
```

in block (1) is used.

To get a rough idea of run time, consider a 50 Hz deep water problem with 60 modes (waterborne modes only) and for 37 radials with 501 range points per radial. On a 1 megaflop workstation, this required about 3 minutes with option 'STD' and 6 hours when including horizontal refraction via option 'GBT'.

Run time is roughly proportional to M * NTHETA * NR.

Dimensional constraints:

Number of modes at a single node <= 1500
Number of distinct sets of modes <= 250

4.4 Plotting routines

4.4.1 PLOTFIELD.HLP

PLOTFIELD produces either rectangular or polar plots of shade files. The former are used for range-depth plots of transmission loss (slices in a vertical plane) and the latter for range-range plots of transmission loss (slices of in a horizontal plane).

```
Files:

        Name            Unit          Description
Input
        *.PLP             1        PLot Parameters
Output
        *.SHD           11-99      SHaDe file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

'RD'                     ! 'S' SCALE, 'L' OR 'D' LINEAR OR DB
'CA'                     ! POLAR OR CARTESIAN ('PO', 'CA')
'VUGC'                   ! DEVICE ('PRX', 'TEK', 'TEK', 'T41')
0                        ! Segment number (0 for none)
0.0                      ! SMOOTHING WINDOW (m)
1                        ! NPLOTS
'SCR:DEEPB'  100.0       ! SOURCE DEPTH
0.0  50.0  0.0  5000.0   ! RMINPL  RMAXPL (km)  ZMIN  ZMAX (m)
 66.0 102.0 4.0          ! TLmin  TLmax (dB)

(1) - SCALING

      Syntax:
         OPT
      Description:
         OPT(1:1): Source type.
                     'R' No scaling applied.
                     'S' Field multiplied by sqrt(r).
         OPT(2:2): Linear or dB
                     'L' Linear scale (useful for pulse plots).
                     'D' dB     scale (for most other plots).


(2) - COORDINATE SYSTEM:
```

```
Syntax:
   COORD
Description:
   COORD(1:2): Source type.
               'CA' Cartesian.
               'PO' Polar.
   COORD(3:3): Grid plotting.
               'G' Grid.
               ' ' No grid.
```

(3) - OUTPUT DEVICE:

```
Syntax:
   DEVICE
Description:
   DEVICE(1:3): Device type.
                'VUG' Vugraph machine.
                'TEK' Tektronix 4691     hardcopy device.
                'THE' Tektronix 4693     hardcopy device.
                'VUG' Tektronix VUGRAPH hardcopy device.
                'T41' Tektronix 41xx terminal.
                'VTT' DEC VT340      terminal.
                'MGP' VAXstation 2000.
                'X11' X-windows device.
                'PSP' PostScript printer.
   DEVICE(4:4): Color or black and white
                'C' Color
                'B' Black and white
   DEVICE(5:5): Smooth (linear interp.) or blocky shading.
                'S' Smooth.
                'B' Blocky.
   DEVICE(6:6): Final copy.
                'F' Final
                ' ' Standard labels
                This option labels the plots (a), (b), (c)
                and strips out certain other information to
                make the plot suitable for a final report.
```

(4) - SEGMENT NUMBER:

```
Syntax:
   SEGNUM
Description:
```

SEGNUM: If you specify a nonzero segment number, then the file will be added into the default segment file. (See UNIRAS documentation for info on segment files).

(5) - SMOOTHING WINDOW

Syntax:

```
DR3DB
```

Description:

DR3DB: Three dB smoothing window (m). The tranmission loss is computed then smoothed in range using a Gaussian filter. DR3DB gives the range interval over which the smoothing is performed.

(6) - NUMBER OF PLOTS

Syntax:

```
NPLOTS
```

Description:

NPLOTS: You can do 1,2 or 3 plots per page as indicated by NPLOTS.

(7) - PLOT DATA:

Syntax:

```
FILNAM  SD
RMIN  RMAX  RINC  RAXL
ZMIN  ZMAX  ZINC  ZAXL
TLMIN TLMAX TLINC
```

Description:

FILNAM:	Name of the shade file. The file name is provided without the extension which is assumed to be '.SHD'.	
SD:	Source depth.	
RMIN:	Minimum range	(km).
RMAX:	Maximum range	(km).
RINC:	Range interval for tick marks	(m/s).
RAXL:	Range axis length	(cm).
ZMIN:	Minimum depth	(m).
ZMAX:	Maximum depth	(m).
ZINC:	Depth interval for tick marks	(m/s).

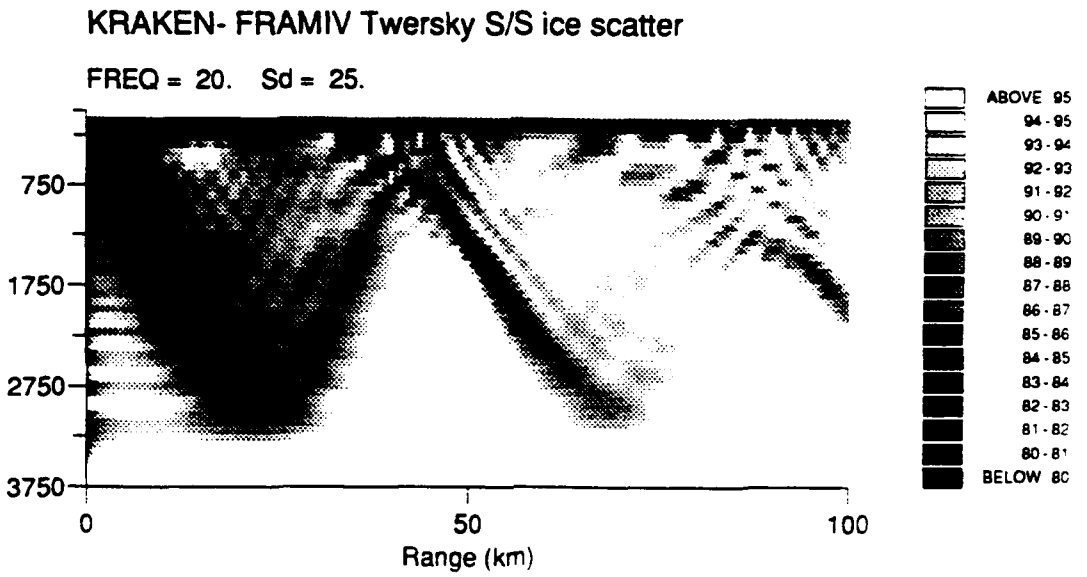


Figure 4.3: Sample output of PLOTFIELD: transmission loss for the Arctic problem.

```
   ZAXL:   Depth axis length              (cm).

   TLMIN:  Minimum transmission loss      (dB).
   TLMAX:  Maximum transmission loss      (dB).
   TLINC:  Increment in transmission loss (dB).

This information should be repeated for ISRC = 1, NSRCS


For a polar plot the form is:

Syntax:
   FILNAM  SD
   XMIN  XMAX  XINC  XAXL
   YMIN  YMAX  YINC  YAXL
   TLMIN TLMAX TLINC

where XMIN, XMAX, YMIN, YMAX, specify the x and y limits of
the plot in kilometers.
```

4.4.2 PLOTGRN.HLP

PLOTGRN uses the modes to produces plots of the amplitude of the Green's function corresponding to a particular source/receiver combination. The Green's function is plotted as a function of slowness (k).

```
Files:

        Name            Unit          Description
Input
        *.PLP             1        PLot Parameters
        *.MOD            30        MODe file
Output
        *.PRT             6        PRinT file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

 25.0  25.0                          ! SD  RD (m)
1400.0  1500.0  2000                 ! CMIN  CMAX  NKPTS
2.5E-6                               ! ATTEN (Stabilizing attenuation)
10.0  8.0                            ! XAXL  YAXL


(1) - SOURCE/RECEIVER DEPTHS

      Syntax:
         SD  RD
      Description:
         SD: Source   depth (m).
         RD: Receiver depth (m).


(2) - PHASE SPEED LIMITS

      Syntax:
         CLOW  CHIGH  NKPTS
      Description:
         CLOW:  Lower phase speed limit (m/s).
         CHIGH: Upper phase speed limit (m/s).
         NKPTS: Number of points in k-space for evaluating
                the kernel.
```

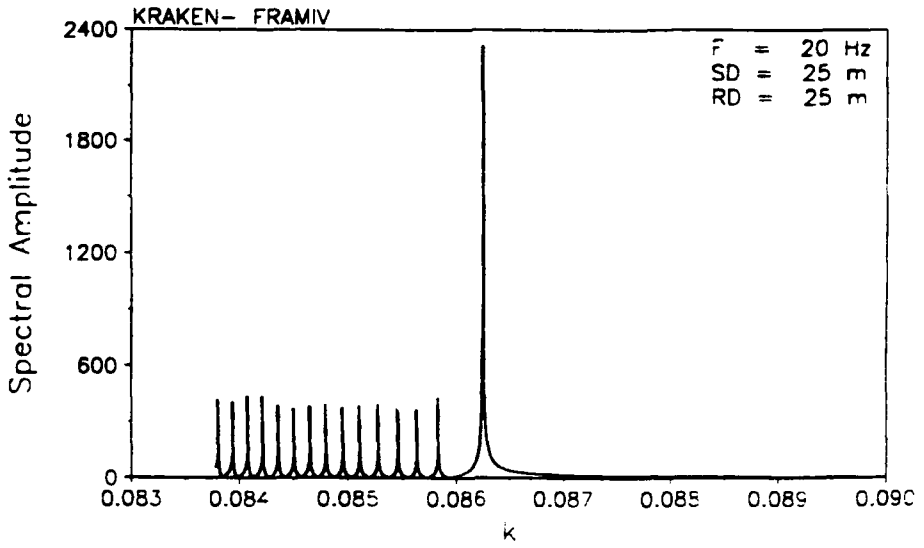


Figure 4.4: Sample output of PLOTGRN: the Green's function for the Arctic problem.

(3) - STABILIZING ATTENUATION

Syntax:
ALPHA
Description:
ALPHA: Stabilizing attenuation (1/m).

Since G(k) has singularities on the real k-axis. it is evaluated on a slice displaced into the complex plane a distance i **ALPHA**.

(4) - AXIS LENGTHS

Syntax:
XAXL YAXL
Description:
XAXL: X-axis length (cm).
YAXL: Y-axis length (cm).

4.4.3 PLOTMODE.HLP

PLOTMODE produces plots of specified modes using a solid line for the real part and and dashed line for the imaginary part. It requires the usual KRAKEN output files containing the modes and an additional file containing a list of which modes to plot terminated by a zero. The number of points used to plot a mode is determined by the number of source/receiver points you used in the original KRAKEN run.

Before plotting each mode the maximum absolute value of the (complex) eigenfunction is calculated, that is,

T = max (abs(Z(z))

where Z(z) denotes the eigenfunction and the maximum is calculated over the depth points in the plot window. The eigenfunctions are then scaled by this factor and plotted along with the value of T. As a result, purely real eigenfunctions will peak at 1 on the plot; complex eigenfunctions may have a peak in their real or imaginary parts which is less than 1.

```
Files:

        Name            Unit         Description
Input
        *.PLP            5       PLot Parameters
        *.MOD           30       MODe file
Output
        *.PRT            6       PRinT file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

'P'
0.0 5000.0 1000.0 15.0    ! ZMIN  ZMAX  ZINC  ZAXL
1
2
3
4
0


(1) - COMPONENT

      Syntax:
         COMP
      Description:
         COMP: Selects which component of a mode is plotted.
               'V' Vertical   displacement
```

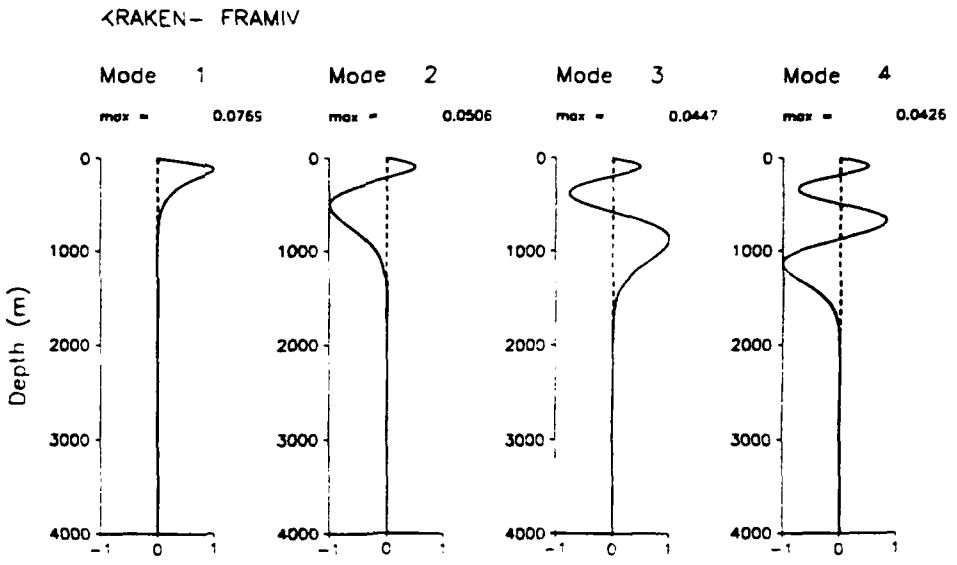


Figure 4.5: Sample output of PLOTMODE: selected modes for the Arctic problem.

```
        'H' Horizontal displacement
        'T' Tangential stress
        'N' Normal     stress
        'P' Pressure (gives same result as 'N')

(2) - DEPTH AXIS INFO

      Syntax:
         ZMIN  ZMAX  ZINC  ZAXL
      Description:
         ZMIN:    Z minimum                  (m ).
         ZMAX:    Z maximum                  (m ).
         ZINC:    Z interval for tick marks  (m ).
         ZAXL:    Z axis length              (cm).
```

4.4.4 PLOTSLICE.HLP

PLOTSLICE produces plots of coherent transmission loss versus range by extracting a particular slice from a shade file which is produced by any of several propagation models including FSTFLD, KRAKEN, SCOOTER and IFD. An additional input file '.PLP' contains plot parameters.

```
Files:

        Name            Unit          Description
Input
        *.PLP             1         PLot Parameters
Output
        *.PRT             6         PRinT file
        *.SHD           11-99       SHaDe file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

'R'                      ! OPT 'R/S' (scaling), 'L/D' (linear/dB)
3                        ! NCURVES
500.0  2500.0            ! SD  RD (m)
200.0   220.0 /          !  RMIN   RMAX   RINC (km)   RAXL (cm)
 70.0   110.0 /          ! TLMIN  TLMAX  TLINC (dB)  TLAXL (cm)
0.0                      ! DR3DB (m) (smoothing window)


(1) - SCALING

      Syntax:
         OPTION
      Description:
         OPTION(1:1): Source type.
                      'R' No scaling applied.
                      'S' Field multiplied by sqrt(r).
         OPTION(2:2): Linear or dB.
                      'L' for linear scaling
                          (useful for pulse plots).
                      'D' for plots in dB
                          (for most other plots).


(2) - NUMBER OF CURVES
```

```
Syntax:
   NCURVES
Description:
   NCURVES:  Number of curves to plot.
             For each curve a shade file name should be
             passed in the parameter list.
```

(3) - SOURCE/RECEIVER DEPTHS

```
Syntax:
   SD  RD
Description:
   SD: Source   depth (m).
   RD: Receiver depth (m).
```

(4) - RANGE AXIS INFO

```
Syntax:
   RMIN  RMAX  RINC  RAXL
Description:
   RMIN:    Range minimum                 (m/s).
   RMAX:    Range maximum                 (m/s).
   RINC:    Range interval for tick marks (m/s).
   RAXL:    Range axis length             (cm).
```

(5) - TRANSMISSION LOSS AXIS INFO

```
Syntax:
   TLMIN  TLMAX  TLINC  TLAXL
Description:
   NCURVES: Number of curves.
   TLMIN:   TL minimum                 (dB).
   TLMAX:   TL maximum                 (dB).
   TLINC:   TL interval for tick marks (dB).
   TLAXL:   TL axis length             (cm).
```

(6) - SMOOTHING:

```
Syntax:
   SIGMA
Description:
```

SIGMA: Smoothing window (m).
A gaussian filter is applied with this
window size.

4.4.5 PLOTSSP.HLP

PLOTSSP produces plots of the sound speed profile. It requires as input an environmental file of exactly the same form used by KRAKEN. It ouputs a print file echoing the input data. The number of points used in plotting the sound speed profile is 200 per medium.

```
Files:

        Name            Unit         Description
Input
        *.ENV             1        ENVironmental data
        *.PLP             3        PLot Parameters
Output
        *.PRT             6        PRinT file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

1425.0  1525.0    25.0  10.0   ! CMIN  CMAX  CINC (m/s)  CAXL (cm)
   0.0  4000.0  1000.0  10.0   ! ZMIN  ZMAX  ZINC (m)    ZAXL (cm)


(1) - SOUND SPEED AXIS INFO

      Syntax:
         CMIN  CMAX  CINC  CAXL
      Description:
         CMIN:  Sound speed minimum                  (m/s).
         CMAX:  Sound speed maximum                  (m/s).
         CINC:  Sound speed interval for tick marks (m/s).
         CAXL:  Sound speed axis length              (cm ).


(2) - DEPTH AXIS INFO

      Syntax:
         ZMIN  ZMAX  ZINC  ZAXL
      Description:
         ZMIN:  Depth minimum                  (m ).
         ZMAX:  Depth maximum                  (m ).
         ZINC:  Depth interval for tick marks (m ).
         ZAXL:  Depth axis length              (cm).
```

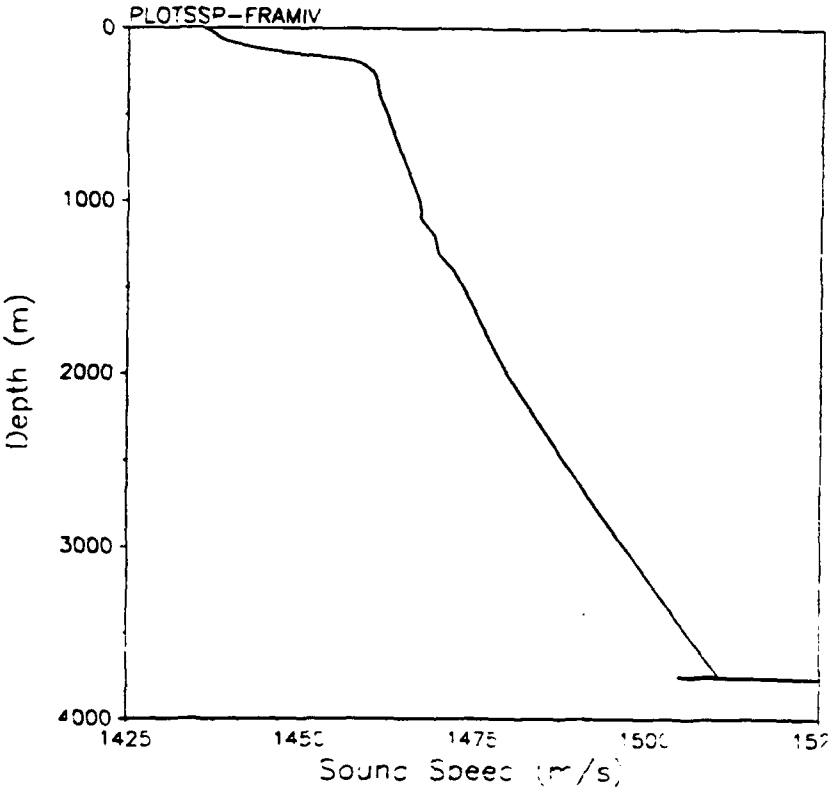


Figure 4.6: Sample output of PLOTSSP: the sound speed profile for the Arctic problem.

4.4.6 PLOTTLD.HLP

PLOTTLD produces plots of coherent transmission loss versus depth. The user must provide a mode file as computed by KRAKEN.

```
Files:

        Name            Unit         Description
Input
        *.PLP             1        PLot Parameters
        *.MOD            30        MODe file
Output
        *.PRT             6        PRinT file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

'R'                          ! OPT (X or R for cart or cyl coords)
999                          ! M   (number of modes to include)
25.0                         ! SD (m)
50.0                         ! Receiver range (km)
501  0.0  3750.0 /           ! NRCVRS  RD(1:NRCVRS) (m)
  0.0 4000.0 1000.0 10.0     !  ZMIN   ZMAX   ZINC  (m)   ZAXL (cm)
 70.0  110.0   10.0 10.0     ! TLMIN  TLMAX  TLINC (dB)  TLAXL (cm)


(1) - OPTIONS

      Syntax:
         OPTION
      Description:
         OPTION(1:1): Source type.
                      'R' for a point source
                      (cylindrical or R-Z coordinates).
                      'X' for a line source
                      (cartesian   or X-Z coordinates)
         OPTION(2:2): Dummy variable for consistency with PLOTTLR.
         OPTION(3:3): Component.
                      ''  (null) for pressure.
                      'H' for Horizontal displacement.
                      'V' for Vertical   displacement.
                      'T' for Tangential stress.
                      'N' for Normal     stress.
```

(2) NUMBER OF MODES

```
Syntax:
   M
Description:
   M: Number of modes to use in the field calculation.
      If this number is larger than the actual number of
      modes in the mode file it is reduced accordingly.
```

(3) - SOURCE/RECEIVER LOCATIONS

```
Syntax:
   SD
   RR
   NRD  RD(1:NRD)
Description:
   SD: Source depth    (m ).
   RR: Receiver range (km).
   NRD:      Number of receivers.
   RD():     Receiver depths (m).
             The format of the source/rcvr depth info is an
             integer indicating the number of receivers
             followed by real numbers indicating the range
             of each receiver.  Since this data is read in
             using list-directed I/O you can type it just
             about any way you want, e.g. on one line or
             split onto several lines.  Also if your depths
             are equally spaced that you can type just the
             first and last depths followed by a '/' and the
             intermediate depths will be generated
             automatically.
```

(6) - Z AXIS INFO

```
Syntax:
   ZMIN  ZMAX  ZINC  ZAXL
Description:
   ZMIN:     Z minimum                  (m ).
   ZMAX:     Z maximum                  (m ).
   ZINC:     Z interval for tick marks  (m ).
   ZAXL:     Z axis length              (cm).
```

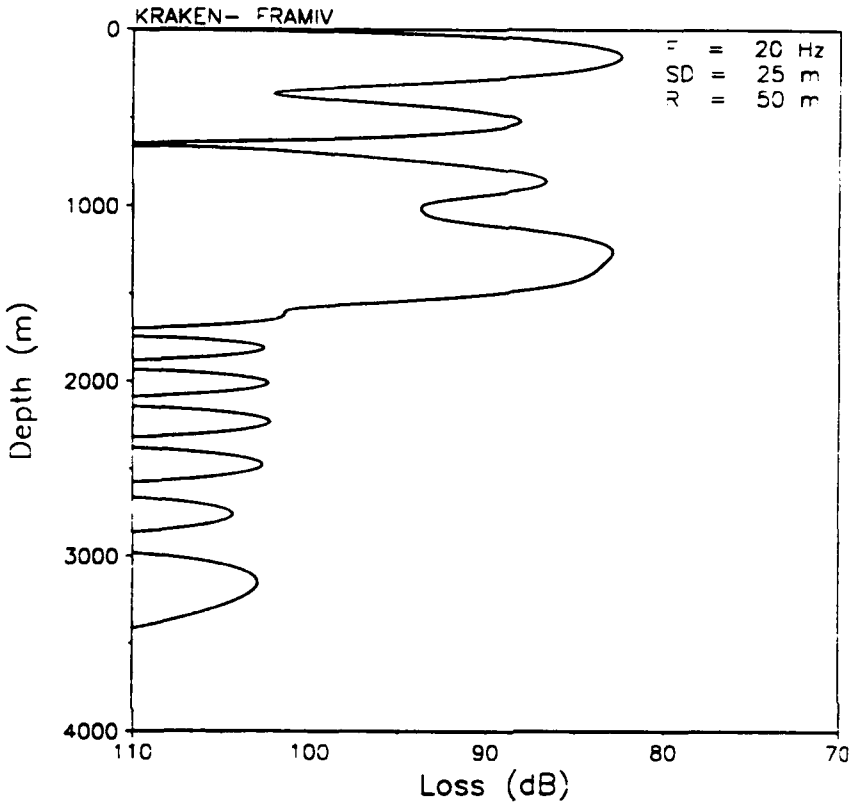


Figure 4.7: Sample output of PLOTTLD: transmission loss vs. depth for the Arctic problem.

```
(7) - TRANSMISSION LOSS AXIS INFO

      Syntax:
         TLMIN  TLMAX  TLINC  TLAXL
      Description:
         TLMIN:    TL minimum                  (dB).
         TLMAX:    TL maximum                  (dB).
         TLINC:    TL interval for tick marks (dB).
         TLAXL:    TL axis length              (cm).

      If you set TLMIN=TLMAX then the curve is autoscaled.
```

4.4.7 PLOTTLR.HLP

PLOTTLR sums the modes to produces plots of coherent transmission loss versus range.

```
Files:

        Name            Unit         Description
Input
        *.PLP             1        PLot Parameters
        *.MOD           30-99      MODe files
Output
        *.PRT             6        PRinT file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

'RAN'                          ! OPT (cart, cylin or scaled coords)
999                            ! M (number of modes to include)
1, 0.0                         ! NPROF  RPROF(1:NPROF) (km)
25.0  25.0                     ! SD  RD (m)
0.0   100.0  501 10.0 10.0     !  RMIN  RMAX  NR  RINC (km)  RAXL (cm)
70.0  110.0      10.0  6.0     ! TLMIN TLMAX     TLINC (dB) TLAXL (cm)
0.0                            ! 3dB smoothing window (m)


(1) - OPTIONS

      Syntax:
         OPTION
      Description:
         OPTION(1:1): Source type.
                      'R' point source.
                          (cylindrical or R-Z coordinates)
                      'X' line source.
                          (cartesian   or X-Z coordinates)
         OPTION(2:2): Selects coupled mode or adiabatic mode theory
                      'C' for Coupled   mode theory.
                      'A' for Adiabatic mode theory (default).
         OPTION(3:3): Component
                      ''  (null) for pressure.
                      'H' for Horizontal displacement.
                      'V' for Vertical   displacement.
                      'T' for Tangential stress.
```

'N' for Normal stress.

(2) - NUMBER OF MODES

Syntax:
M
Description:
M: Number of modes to use in the field computation.
If this number is larger than the actual number of modes in the mode file it is reduced accordingly.

(3) - PROFILE RANGES

Syntax:
NPROF RPROF(1:NPROF)
Description:
NPROF: The number of profiles, i.e. ranges where a new set of modes is to be used.
RPROF(): Ranges (km) of each of these profiles.
For a range independent problem there is only one profile and its range is arbitrary. Mode files must exist for each range of a new profile and be assigned in sequence to units 30,31,... The modes for the last SSP profile are extended in a range-independent fashion to infinity so that RMAX can exceed RPROF(NPROF).

(4) - SOURCE/RECEIVER DEPTHS

Syntax:
SD RD
Description:
SD: Source depth (m).
RD: Receiver depth (m).

(5) - RECEIVER RANGES/AXIS INFO

Syntax:
RMIN RMAX NR RINC RAXL
Description:
RMIN: First receiver range (km).

```
   RD:   Last  receiver range          (km).
   NR:   Number of receiver ranges.
   RINC: Range interval for tick marks (km).
   RAXL: Range axis length             (cm).
```

(6) - TRANSMISSION LOSS AXIS INFO

```
Syntax:
   TLMIN  TLMAX  TLINC  TLAXL
Description:
   TLMIN:    TL minimum                 (dB).
   TLMAX:    TL maximum                 (dB).
   TLINC:    TL interval for tick marks (dB).
   TLAXL:    TL axis length             (cm).
```

If you set TLMIN=TLMAX then the curve is autoscaled.

(7) - SMOOTHING WINDOW

```
Syntax:
   DR3DB
Description:
   DR3DB: Three dB smoothing window (m).
          The tranmission loss is computed then smoothed in
          range using a Gaussian filter. DR3DB gives the
          range interval over which the smoothing is
          performed.
```

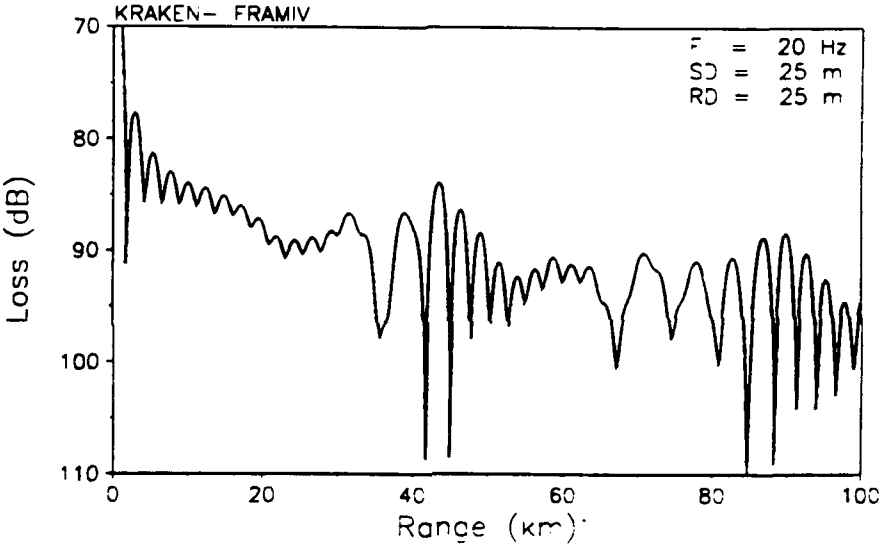


Figure 4.8: Sample output of PLOTTLR: transmission loss vs. range for the Arctic problem.

4.4.8 PLOTTRI.HLP

PLOTTRI takes an input file '.FLP' in the same format that FIELD3D uses. and produces a plot showing the layout of triangular patches which the user has specified.

```
Files:

        Name            Unit          Description
Input
        *.PLP             1        PLot Parameters
        *.FLP            20        FieLd Parameters
Output
        *.PRT             6        PRinT file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

-200.0  100.0  0.0  10.0 /     ! XMIN  XMAX  XINC (km)  XAXL (cm)
-100.0  200.0  0.0  10.0 /     ! YMIN  YMAX  YINC (km)  YAXL (cm)

(1) - X-AXIS INFO

      Syntax:
         XMIN  XMAX  XINC  XAXL
      Description:
         YMIN: The minimum Y-value          (km).
         YMIN: The maximum Y-value          (km).
         YINC: Y-interval for tick marks    (km).
         YAXL: Y-axis length                (cm).


(2) - Y-AXIS INFO

      Syntax:
         YMIN  YMAX  YINC  YAXL
      Description:
         YMIN: The minimum Y-value          (km).
         YMIN: The maximum Y-value          (km).
         YINC: Y-interval for tick marks    (km).
         YAXL: Y-axis length                (cm).
```

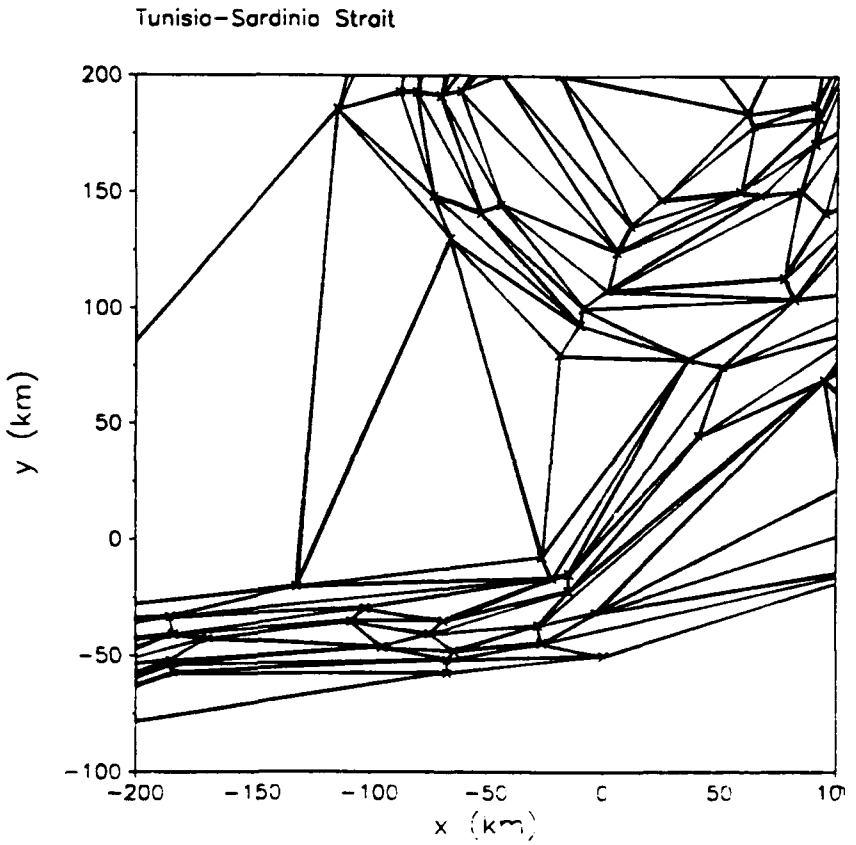


Figure 4.9: Sample output of PLOTTRI: triangulation used for a 3D Mediterranean scenario.

4.5 The BELLHOP ray/beam model

It is often useful to be able to plot the rays for a given environment for illustrative purposes. Also for high-frequency problems the ray model provides results more rapidly. The results are almost inevitably less accurate than KRAKEN or SCOOTER calculations but sometimes the inaccuracy is negligible.

BELLHOP is not particularly efficient as far as ray models are concerned. nor is it very general. With regard to efficiency the model actually is set up internally to do a full range-dependent ray trace while it allows only for a range-independent input structure.

BELLHOP differs from standard ray models in using a robust variant of Gaussian beam tracing referred to as geometric beam tracing. For those familiar with traditional implementations of ray tracing, the results are often astonishingly accurate.

4.5.1 BELLHOP.HLP

BELLHOP computes acoustic fields in oceanic environments via Gaussian beam tracing. The environment treated consists of an acoustic medium with a sound speed which may depend on range and depth. A theoretical description may be found in:

Michael B. Porter and Homer P. Bucker, "Gaussian beam tracing for computing ocean acoustic fields," J. Acoust. Soc. Amer. 82, 1349–1359 (1987).

The following programs are used with BELLHOP:

BELLHOP Main program for doing Gaussian beam tracing

PLOTRAY Produces plots of central rays of beams

ANGLES Given the source and recvr sound speeds, computes the angle of the limiting ray.

PLOTSSP Plots the sound speed profile

BELLHOP produces pressure fields in the NRL standard format. These fields can then be plotted using the following routines:

PLOTSLICE Plots a transmission loss versus range curve.

PLOTFIELD Plots a full TL field versus range and depth.

The steps in running the program are as follows:

1. Set up your environmental file and run PLOTSSP to make sure the SSP looks reasonable.

2. Do a ray trace. That is,

   A. Run BELLHOP with the ray trace option to calculate about 50 rays.

   B. Run PLOTRAY to make sure you have the angular coverage you expect. Do the rays behave irregularly? If so reduce the step-size and try again.

3. Re-run BELLHOP using the coherent, incoherent or semicoherent option for transmission loss. (Use the default number of beams.)

4. Run either PLOTSLICE to plot a single transmission loss curve or PLOTFIELD to plot a full range-depth field plot.

5. Double the number of beams and check convergence.

Files:

	Name	Unit	Description
Input			
	*.ENV	1	ENVironmental data
Output			
	*.PRT	6	PRinT file
	*.RAY	21	RAY file
	*.SHD	25	SHaDe file

EXAMPLE AND DESCRIPTION OF ENV FILE:

```
'Munk profile'          ! TITLE
50.0                    ! FREQ (Hz)
1                       ! NMEDIA
'SVN'                   ! SSPOPT (Analytic or C-linear interpolation)
51  0.0  5000.0        ! DEPTH of bottom (m)
    0.0  1548.52  /
  200.0  1530.29  /
  250.0  1526.69  /
  400.0  1517.78  /
  600.0  1509.49  /
  800.0  1504.30  /
 1000.0  1501.38  /
 1200.0  1500.14  /
 1400.0  1500.12  /
 1600.0  1501.02  /
 1800.0  1502.57  /
 2000.0  1504.62  /
 2200.0  1507.02  /
```

```
 2400.0  1509.69  /
 2600.0  1512.55  /
 2800.0  1515.56  /
 3000.0  1518.67  /
 3200.0  1521.85  /
 3400.0  1525.10  /
 3600.0  1528.38  /
 3800.0  1531.70  /
 4000.0  1535.04  /
 4200.0  1538.39  /
 4400.0  1541.76  /
 4600.0  1545.14  /
 4800.0  1548.52  /
 5000.0  1551.91  /
'V'  0.0
1  1000.0 /                  ! NSD  SD(1:NSD) (m)
2    0.0 5000.0 /             ! NRD  RD(1:NRD) (m)
501  0.0  100.0 /             ! NRR  RR(1:NR ) (km)
'R'                           ! Run-type: 'R/C/I/S'
51 -11.0 11.0 /               ! NBEAMS  ALPHA(1:NBEAMS) (degrees)
200.0  5500.0  101.0          ! STEP (m)  ZBOX (m)  RBOX (km)
```

DESCRIPTION OF INPUTS:

(1) - TITLE

Syntax: TITLE

Description:
TITLE: Title of run enclosed in single quotes

(2) - FREQUENCY

Syntax: FREQ

Description:
FREQ: Frequency in Hz

(3) - NUMBER OF MEDIA

Syntax: NMEDIA (<20)

```
Description:
   Dummy parameter for compatibility with KRAKEN.

(4) - OPTIONS

Syntax: OPTION

Description:
   OPTION(1:1): Type of interpolation to be used for the SSP
                'S' for cubic Spline  (recommended)
                'C' for C-linear
                'N' for N2-linear
                'A' for Analytic.  The user must modify the
                    analytic formulas in ANALYT.FOR and re-link.

                Use PLOTSSP to check that
                the SSP looks the way you thought it should.
                Apart from potential typos, this will also
                show up fit-problems which might occur with
                the spline option.  Splines yield a
                poor fit to certain kinds of curves, e.g.
                curves with sharp bends.

   OPTION(2:2): Type of top boundary condition
                'V' VACUUM above top
                'R' Perfectly RIGID

                 For open ocean problems option 'V' should be
                 used for the surface BC.

   OPTION(3:3): Volume attenuation option
                'T' Thorp attenuation formula.
                'N' No volume attenuation.

(5) - SOUND SPEED PROFILE

Syntax:
   NMESH  SIGMA  Z(NSSP)
   Z(1)     CP(1) /
   Z(2)     CP(2) /
   .
   .
   .
   Z(NSSP)  CP(NSSP) /
```

```
Description:

   NMESH:   Dummy parameter for KRAKEN compatibility
   SIGMA:   Dummy parameter for KRAKEN compatibility
   Z(NSSP): Depth at bottom of medium (m).
            This value is used to detect the last
            SSP point when reading in the
            profile which follows.

         The following should be omitted when the 'A' option
         is used (indicating that an analytic profile is
         supplied by a user written subroutine).

   Z():     Depth (m).  Note that the surface starts at the first
            depth point specified. Thus if you have say, XBT
            data which starts at 50 m below the surface, then
            you'll need to put in some SSP point at 0 m,
            otherwise the free-surface would be placed at 50 m
            giving erroneous results. Try to keep the number of
            depth points to the minimum necessary to describe the
            physics: a fine SSP sampling can force a fine step-size
            for integrating the rays.
   CP():    P-wave speed (m/s) (Must be followed by a '/'
            for compatibility with the KRAKEN program.)

(6) - OPTIONS

      Syntax: OPTION  SIGMA

      Description:
         OPTION(1:1): Type of bottom boundary condition
                      'V' VACUUM below bottom
                      'R' Perfectly RIGID

         SIGMA:        Bottom roughness (currently ignored)

(7) - SOURCE/RECEIVER DEPTHS AND RANGES

      Syntax:
         NSD  SD(1:NSD)
         NRD  RD(1:NRD)
         NR    R(1:NR )

      Description:
```

```
   NSD:  The number of source depths (<51)
   SD(): The source depths (m)
   NRD:  The number of receiver depths (<101 and NR*NRD <= 52000)
   RD(): The receiver depths (m)
   NR:   The number of receiver ranges
         (NR < 1001 and NR*NRD <= 50000)
   R():  The receiver ranges (km)
```

This data is read in using list-directed I/O you can type it just about any way you want, e.g. on one line or split onto several lines. Also if the depths or ranges are equally spaced then you can type just the first and last depths followed by a '/' and the intermediate depths will be generated automatically.

(8) - RUN TYPE

Syntax:

```
   OPTION
```

Description:

```
   OPTION: 'R' generates a ray file
           'C' for     Coherent TL calculation
           'I' for   Incoherent TL calculation
           'S' for Semicoherent TL calculation
               (Lloyd mirror source pattern)
```

(9) - BEAM FAN

Syntax:

```
   NBEAMS ALPHA(1:NBEAMS)
```

Description:

```
   NBEAMS: Number of beams (use 0 to have the program
           calculate a value automatically.
   ALPHA(): Beam angles (negative angles toward surface)
```

For a ray trace you can type in a sequence of angles or you can type the first and last angles followed by a '/'. For a TL calculation, the rays must be equally spaced otherwise the results will be incorrect.

(10) - NUMERICAL INTEGRATOR INFO

Syntax:

STEP ZBOX RBOX

Description:

STEP: The step size used for tracing the rays (m).
ZBOX: The maximum depth to trace a ray (m).
RBOX: The maximum range to trace a ray (km).

The required step size depends on many factors. This includes frequency, size of features in the SSP (such as surface ducts), range of rcvrs, and whether a coherent or incoherent TL calculation is performed. If you use STEP=0.0 BELLHOP will use a default step-size and tell you what it picked. You should then halve the step size until the results are convergent to your required accuracy. To obtain a smooth ray trace you should use the spline SSP interpolation and a step-size less than the smallest distance between SSP data points.

Rays are traced until they exit the box (ZBOX, RBOX). By setting ZBOX less than the water depth you can eliminate bottom reflections.

4.5.2 PLOTRAY.HLP

PLOTRAY produces plots of the rays generated by BELLHOP and contained in a file 'P1'.RAY.

```
Files:

        Name            Unit         Description
Input
        *.PLP             1       PLot Parameters
        *.RAY             9       RAY   file

-----------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

0.0  2500.0    500.0   5.0/        ! ZMIN  ZMAX  ZINC (m)   ZAXL (cm)
0.0    75.0     25.0  10.0/        ! RMIN  RMAX  RINC (km)  RAXL (cm)
'RED' 'GREEN' 'YELLOW' 'YELLOW'
'DASH' 'DOT' 'SOLID' 'SOLID'

(1) - DEPTH AXIS INFO

      Syntax:
         ZMIN  ZMAX  ZINC  ZAXL
      Description:
         ZMIN:  Depth minimum                 (m/s).
         ZMAX:  Depth maximum                 (m/s).
         ZINC:  Depth interval for tick marks (m/s).
         ZAXL:  Depth axis length             (cm ).


(2) - RANGE AXIS INFO

      Syntax:
         RMIN  RMAX  RINC  RAXL
      Description:
         RMIN:  Range minimum                 (km).
         RMAX:  Range maximum                 (km).
         RINC:  Range interval for tick marks (km).
         RAXL:  Range axis length             (cm).

(3) - RAY COLORS

      Syntax:
```

```
      COLR  COLS  COLB  COLSB
   Description:
      COLR:  Color for paths which are purely Refracted.
      COLS:  Color for paths which strike the Surface only.
      COLB:  Color for paths which strike the Bottom  only.
      COLSB: Color for paths which strike both Surface and Bottom.

      Choose from 'BLACK', 'WHITE', 'RED', 'GREEN', 'CYAN',
      'MAGENTA', 'YELLOW'.
      (default is 'BLACK' for all rays)

(4) - RAY PATTERNS

   Syntax:
      PATR  PATS  PATB  PATSB
   Description:
      PATR:  Pattern for paths which are purely Refracted.
      PATS:  Pattern for paths which strike the Surface only.
      PATB:  Pattern for paths which strike the Bottom  only.
      PATSB: Pattern for paths which strike both Surface and Bottom.

      Choose from 'SOLID', 'DASH', 'DOT'.
      (default is 'SOLID' for all rays)
```

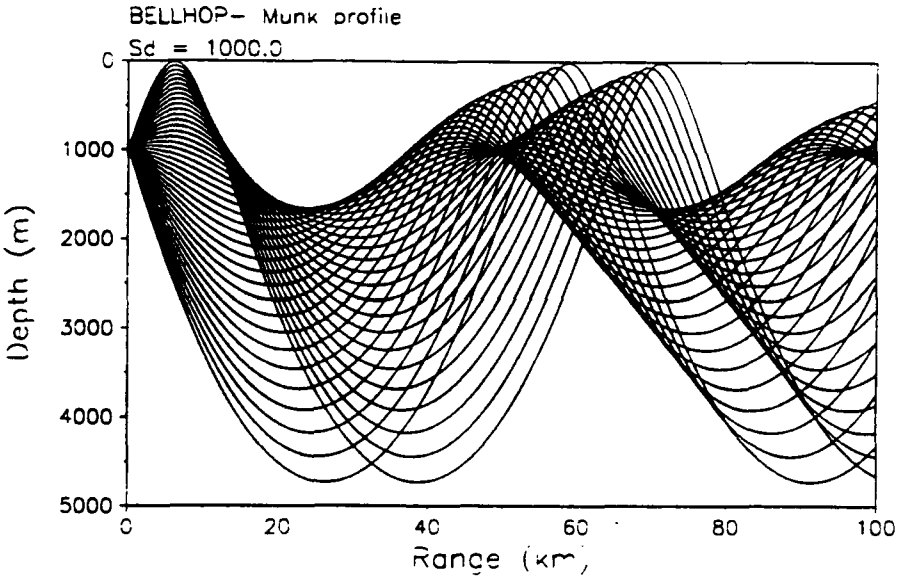


Figure 4.10: Sample output of PLOTRAY: ray trace for the Munk profile

4.6 The SCOOTER FFP model

Both normal mode models and Fast-Field Programs are based on a contour integral representation of the acoustic pressure. Normal mode models evaluate the integral by residues which involves finding the poles of the Green's function. FFP models evaluate the integral directly by stepping along the contour.

In terms of efficiency, the question is whether one can find poles more rapidly than directly integrating. It takes about 15 Green's funtion evaluations to find a pole. The number of Green's function evaluations for an FFP model increases linearly with range.

On the other hand, the normal mode series neglects certain contributions which tend to be important in the near-field (say within 10 water depths). Indeed some problems have no modes at all such as the problem of a point source in free space. Also for very complicated problems (with elasticity) it can be difficult to reliably find the modes. Then an FFP model is a good alternative.

A complete explanation of when to use which model would require many pages. A rule-of-thumb is to use SCOOTER when you are concerned about the field within 10 water depths and KRAKEN otherwise. SCOOTER can be used for larger ranges but will generally require more CPU time. KRAKEN can be run for closer ranges but requires some insight in setting up the environment to make sure that the modes are adequate for describing the field. This is done either by extending the model of the ocean bottom in depth, introducing a false bottom, or computing leaky modes.

4.6.1 SCOOTER.HLP

SCOOTER is a finite element code for computing acoustic fields in range-independent environments. The method is based on direct computation of the spectral integral (reflectivity or FFP method). Pressure is approximated by piecewise-linear elements as are the material properties. (One exception is the density which is approximated by piecewise constant elements).

The SCOOTER package includes two modules:

```
SCOOTER the main program

FIELDS  Produces shade files or
        plots of the Green's function and transmission loss.
```

```
The input (.ENV) file is identical to that used by KRAKEN or KRAKENC.
The output is a Green's function file  (in place of the mode file
produced by KRAKEN ).

Note that SCOOTER includes the effect of density gradients within
media (KRAKEN and KRAKENC do not). Also, interfacial scatter is not
treated in SCOOTER.
```

```
Files:

        Name            Unit        Description
Input
        *.ENV            1        ENVironmental data
        *.BRC           10        Bottom   Refl. Coef.  (optl)
        *.TRC           11        Top      Refl. Coef.  (optl)
        *.IRC           12        Internal Refl. Coef.  (optl)
Output
        *.PRT            6        PRinT file
        *.GRN           20        GReen's function

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF ENV FILE:

'Pekeris problem'
10.0
1
'NVF'
500  0.0  2
     0.0  1500.0 /
  5000.0  1500.0 /
'A'  0.0
  5000.0  2000.0  0.0  2.0 /
1400.0  2000.0
500.0                    ! RMAX (km)
1 500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /               ! NRD  RD(1:NRD)
```

RMAX is the maximum range for a receiver. It translates directly into the number of k-space points that will be used in the spectral integral. CPU time is proportional to RMAX so it shouldn't be any larger than necessary.

Note that both source and receiver must lie within the finite element domain. That is, the capability for placing source or receiver in the homogeneous half-space has not been implemented.

CPU time is roughly independent of the number of receivers but increases linearly with the number of sources. (However, the first source requires about 3 times as much CPU time as subsequent sources, since an LU decomposition is required only for the first source.)

Shade files or plots of transmission loss versus range are obtained by running FIELDS which uses the '.GRN' file as input.

4.6.2 FIELDS.HLP

The FIELDS program uses the Green's functions calculated by SCOOTER or SPARC and produces a shade file that contains a sequence of snapshots of the acoustic field as a function of range and depth. Alternatively, if a single source/receiver combination is specified then FIELDS produces a plot directly of the Green's function and transmission loss.

```
Files:

        Name            Unit         Description
Input
        *.PLP             1        PLot Parameters
        *.GRN            20        GReen's function
Output
        *.PRT             6        PRinT file
        *.SHD            25        SHaDe file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF ENV FILE:

'RDB'                          ! 'R/X (coord), Lin/DB, Pos/Neg/Both'
200.0  220.0  501              ! RMIN, RMAX, NR
0.0  0.0                       ! SOURCE, RCVR DEPTHS (0,0 FOR ALL)
70.0  110.0                    ! TLMIN, TLMAX

 (1) - OPTIONS

      Syntax:
         OPT
      Description:
         OPT(1:1): Coordinates
                   'R' Cylindrical (R-Z) coordinates.
                   'X' Cartesian   (X-Z) coordinates.
         OPT(2:2): Scale
                   'D' dB
                   'L' linear
         OPT(3:3): Spectrum
                    'P' Positive (recommended)
                    'N' Negative
                    'B' Both positive and negative
                   The spectral integral should formally be done
                   from all along the real k-axis, however the
                   negative portion contributes significantly
```

only in the near-field. Run-time is less
if it is neglected.

(2) - RECEIVER RANGES

Syntax: RMIN RMAX NR

Description:
RMIN: First receiver range (km)
RD: Last receiver range (km)
NR: Number of receiver ranges

(3) - SOURCE/RECEIVER DEPTHS

Syntax: SD RD

Description:
SD: Source depth (m)
RD: Receiver depth (m)

Specify zero SD and RD to have the entire file converted.

(4) - TRANSMISSION LOSS AXIS INFO

Syntax: TLMIN TLMAX TLINC TLAXL

Description:

TLMIN:	Transmission loss minimum	(dB)
TLMAX:	Transmission loss maximum	(dB)
TLINC:	Transmission loss interval for tick marks	(dB)
TLAXL:	Transmission loss axis length	(cm)

These lines are ingnored if you are running FIELDS to simply produce a Green's function file and not a plot.

4.7 The SPARC pulse model

4.7.1 SPARC.HLP

SPARC (SACLANTCEN Pulse Acoustic Research Code) is an experimental time-marched FFP. It treats problems with broadband or transient sources, that is, pulses. The environmental file is patterned after that used for KRAKEN and SCOOTER. The mathematical basis and numerical algorithm is described in:

Michael B. Porter, "The Time-Marched FFP for Modeling Acoustic Pulse Propagation," J. Acoust. Soc. Amer. 87, 2013–2023 (1990).

```
Files:

        Name            Unit        Description
Input
        *.ENV             1       ENVironmental data
        *.STS            10       Source Time Series
Output
        *.PRT             6       PRinT file
        *.GRN            20       GReen's function
        *.RTS            35       Receiver Time Series

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF ENV FILE:

'Munk profile'
5.0
2
'NVWS'
500  0.0  27
    0.0  1548.52  0.0  1.0  0.0  0.0
  200.0  1530.29 /
  250.0  1526.69 /
  400.0  1517.78 /
  600.0  1509.49 /
  800.0  1504.30 /
 1000.0  1501.38 /
 1200.0  1500.14 /
 1400.0  1500.12 /
 1600.0  1501.02 /
 1800.0  1502.57 /
 2000.0  1504.62 /
 2200.0  1507.02 /
 2400.0  1509.69 /
```

```
 2600.0  1512.55 /
 2800.0  1515.56 /
 3000.0  1518.67 /
 3200.0  1521.85 /
 3400.0  1525.10 /
 3600.0  1528.38 /
 3800.0  1531.70 /
 4000.0  1535.04 /
 4200.0  1538.39 /
 4400.0  1541.76 /
 4600.0  1545.14 /
 4800.0  1548.52 /
 5000.0  1551.91 /
500  0.0  2
  5000.0  1551.91  0.0  1.00  1.0 0.0
 10000.0 /
'R'  0.0
1500.0  1550.0
10.0                          ! RMAX (km)
1   250.0 /                   ! NSD  SD(1:NSD) (m)
26  0.0 5000.0 /              ! NRD  RD(1:NRD) (m)
'PH'  0.0  15.0               ! PULSE  FMIN  FMAX (Hz)
1  60.0  0.200 /              ! NRR  RR(1:NRR) (km)
6 1.0 3.0 5.0 10.0 20.0 30.0  ! NTOUT  TOUT(1:NTOUT)
-0.1  0.9  0.0  0.0  0.0      ! TSTART (s)  TMULT  ALPHA  BETA  V (m/s)
```

The input structure is identical to KRAKEN except for additional option in line 4 and 4 additional lines at the end.

OPT(4:4): Type of calculation

'S' for Snapshot.
FIELDS must be run afterwards to convert the '.GRN' file to a '.SHD' file containing the pressure field. The shade file can then be plotted using PLOTFIELD.

'R' for Range stack (horizontal array).
The time series is written in a '.RTS' (Receiver Time Series) file which can be plotted using PLOTTS

'D' for Depth stack (vertical array).
The time series is also plotted using PLOTTS.

```
Additional lines:

(1) - SOURCE PULSE INFORMATION:

      Syntax: PULSE  FMIN  FMAX

      Description:
         PULSE(1:1): Type of interpolation to be used for the SSP
            'P' Pseudo-Gaussian
            'R' Ricker wavelet
            'A' Approximate Ricker wavelet
            'S' Single sine
            'H' Hanning weighted four sine
            'N' N-wave
            'G' Gaussian
            'F' From a '.STS' (Source Time Series) file.
            'B' From a '.STS' file Backwards
         PULSE(2:2): Hilbert transforming.
            'H' perform a Hilbert transform of the source
            'N' don't
             Hilbert transforming is used to eliminate
             the left travelling wave.
         PULSE(3:3): Source sign flipping.
            '+' don't flip it (recommended)
            '-' flip it
         PULSE(4:4): Source filtering.
            'L' low cut filter
            'H' high cut filter
            'B' both high and low cut filter
            'N' no cut
         FMIN:  Low  cut frequency (Hz)
         FMAX:  High cut frequency (Hz).
                This should be no higher than necessary
                since the CPU costs are
                proportional to the bandwidth.


(3) - RECEIVER RANGES

      Syntax: NRR  RR(1:NRR)

      Description:
         NRR:  Number of receiver ranges
         RR(): Receiver ranges (km)
```

```
      This line is ignored unless option 'R' has been selected for a
      range-stack.


(3) - OUTPUT TIMES

      Syntax: NTOUT  TOUT(1:NTOUT)

      Description:
         NTOUT:  Number of output times
         TOUT(): Output times (s)


(3) - TIME INTEGRATION PARAMETERS

      Syntax: TSTART  TMULT  ALPHA  BETA V

      Description:
         TSTART:  Starting time for the march.  This should always be
                  earlier than the time at which the source begins to
                  rise.
         TMULT:   Time step multiplier. Specifying TMULT = 1.0 means
                  that the maximum stable time step is used.
         ALPHA:   Lumping parameter
         BETA:    Explicitness parameter
         V:       Convection velocity
```

A good check of convergence can be done by running an isovelocity problem with a gaussian pulse. The pulse should, of course, be undistorted at the receivers. Remember that Hilbert transforming the source causes it to rise early so TSTART has to be adjusted accordingly. It's a good habit to plot the source function using PLOTTS before running SPARC.

4.7.2 PLOTTS.HLP

PLOTTS PLOTs the Time Series used in or created by a SPARC .

```
Files:

        Name            Unit        Description
Input
        *.PLP            1        PLot Parameters
        *.?TS           10        Source/Receiver Time Series
Output
        *.PRT            6        PRinT file
        *.GRN           20        GReen's function
        *.RTS           35        Receiver Time Series

-------------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

 5.0  0.0  5000.0  'PH+N'        ! FREQ  FMIN  FMAX  PULSE
 0.0  1.5  0.5 /                ! TMIN  TMAX  TINC (s)  TAXL (cm)
-1.0  1.0  1.0 /                ! YMIN  YMAX  YINC (m)  YAXL (cm)

(1) - FREQUENCY SPECTRUM / PULSE INFORMATION:

      Syntax:
         FREQ  FMIN  FMAX  PULSE
      Description:
         FREQ:       Characteristic frequency (Hz).
                     This is only used for those canned signals
                     defined by a characteristic frequency.
         FMIN:       Low  cut frequency       (Hz)
         FMAX:       High cut frequency       (Hz).
         PULSE(1:1): Signal type
                     'P' Pseudo-Gaussian
                     'R' Ricker wavelet
                     'A' Approximate Ricker wavelet
                     'S' Single sine
                     'H' Hanning weighted four sine
                     'N' N-wave
                     'G' Gaussian
                     'F' From an '.STS' (Source Time Series) file.
                     'B' From an '.STS' file Backwards
         PULSE(2:2): Hilbert transforming.  This is used to eliminate
                     the left travelling wave.
```

```
                'H' perform a Hilbert transform of the signal
                'N' don't
        PULSE(3:3): Signal sign flipping.
                '+' don't flip it
                '-' flip it
        PULSE(4:4): signal filtering.
                'L' low cut filter
                'H' high cut filter
                'B' both high and low cut filter
                'N' no cut
```

(2) - TIME AXIS INFO

```
Syntax:
   TMIN  TMAX  TINC  TAXL
Description:
   TMIN: First time                    (s ).
   TMAX: Last  time                    (s ).
   TINC: Time interval for tick marks  (s ).
   TAXL: Time axis length              (cm).
```

(3) - VERTICAL AXIS INFO

```
Syntax:
   YMIN  YMAX  YINC  YAXL
Description:
   YMIN:    Y minimum                  (m ).
   YMAX:    Y maximum                  (m ).
   YINC:    Y interval for tick marks  (m ).
   YAXL:    Y axis length              (cm).
```

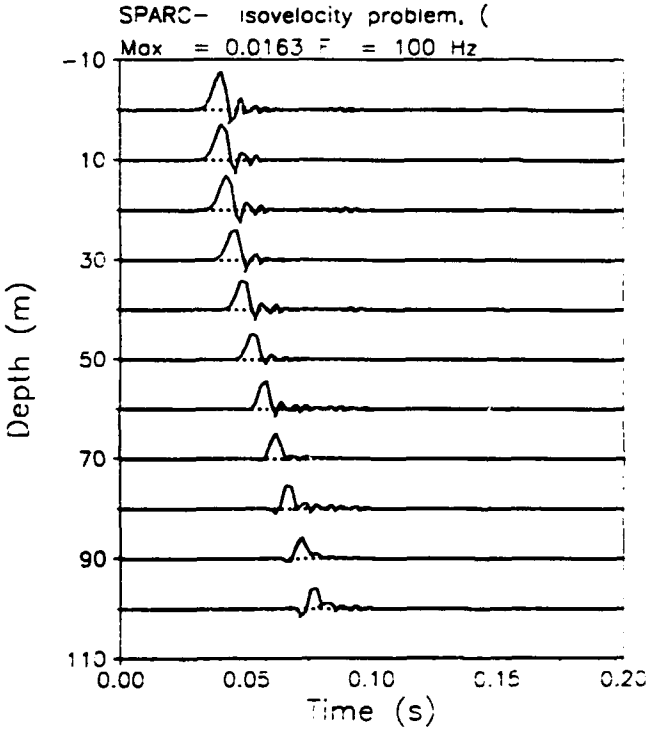


Figure 4.11: Sample output of PLOTTS: an isovelocity problem.

4.8 The BOUNCE reflection coefficient model

4.8.1 BOUNCE.HLP

BOUNCE computes the reflection coefficient for a stack of acoustic media optionally overlying elastic media. The reflection coefficient is written to a '.IRC' file (internal reflection coefficient). This file can be used by KRAKENC to provide a boundary condition or plotted using PLOTRTH.

The input structure is identical to that used by KRAKENC although the input lines for source and receiver depth are not read and can be omitted. Furthermore, the surface boundary condition is ignored and, in effect, replaced by a homogeneous halfspace where the incident wave propagates.

If you are interested in getting a reflection coefficient for a bottom which is being used in a KRAKENC run you will need to delete the layers corresponding to the water column. Otherwise you will get a reflection coefficient corresponding to a wave incident from above the ocean surface.

The angles used for calculating the reflection coefficient are calculated based on the phase velocity interval [CMIN, CMAX]. For a full 90 degree calculation set CMIN to the lowest speed in the problem (say 1400.0) CMAX to 1.0E9. The actual number of tabulated points is determined by RMAX.

I suggest you pick RMAX equal to 10 km, interrupt BOUNCE after about 5 seconds and look at NKPTS which is displayed in the print file. You can then increase or decrease RMAX to obtain adequate sampling of the reflection loss curve (200 points is probably sufficient).

```
Files:

        Name            Unit        Description
Input
        *.ENV            1       ENVironmental data
        *.BRC           10       Bottom   Refl. Coef.   (optl)
        *.IRC           12       Internal Refl. Coef.   (optl)

Output
        *.PRT            6       PRinT file

-------------------------------------------------------

EXAMPLE OF ENV FILE:

'Refl. coef. test problem'
50.0
1
'NVW'
100 0.0 20.0
```

```
    0.0 1600.0 400.0 1.8 0.2 0.5
   20.0 /
'A' 0.0
   20.0 1800.0 600.0 2.0 0.1 0.2
1400.0 19000.0
10.0                    ! RMAX (km)
1 50.0 /                ! NSD  SD(1:NSD)
501 0.0 150.0 /         ! NRD  RD(1:NRD)
```

The above example (taken from the SAFARI reference manual) involves two elastic layers.

4.8.2 PLOTRTH.HLP

PLOTRTH produces plots of the plane wave reflection coefficient as a function of angle (R(THETA)). Reflection loss defined as -20 log10 (R) may also be plotted. An Internal Reflection Coefficient file computed by BOUNCE provides the input.

```
Files:

        Name            Unit         Description
Input
        *.PLP             1        PLot Parameters
        *.IRC            10        Internal Refl. Coef.
Output
        *.PRT             6        PRinT file

-------------------------------------------------------

EXAMPLE AND DESCRIPTION OF PLP FILE:

'D',                          ! 'L/D' linear or dB scale
1500.0,                       ! C0
0.0  90.0  10.0 /             ! THMIN, THMAX, THINC, THAXL (cm)
0.0  15.0   5.0 /             ! RMIN,  RMAX,  RINC,  RAXL  (cm)

(1) - OPTIONS

      Syntax:
         OPTION
      Description:
         OPTION(1:1): Reflection coef. or reflection loss.
                      'L' Linear scale
                          (reflection coef.)
                      'D' dB
                          (reflection loss.)

(2) - Reference wave speed

      Syntax:
         C0
      Description:
         C0: Sound speed in the halfspace.
             The reflection coefficient depends on the sound speed
             and density of the medium from which the wave is incident.
             The density is assumed unity.
```

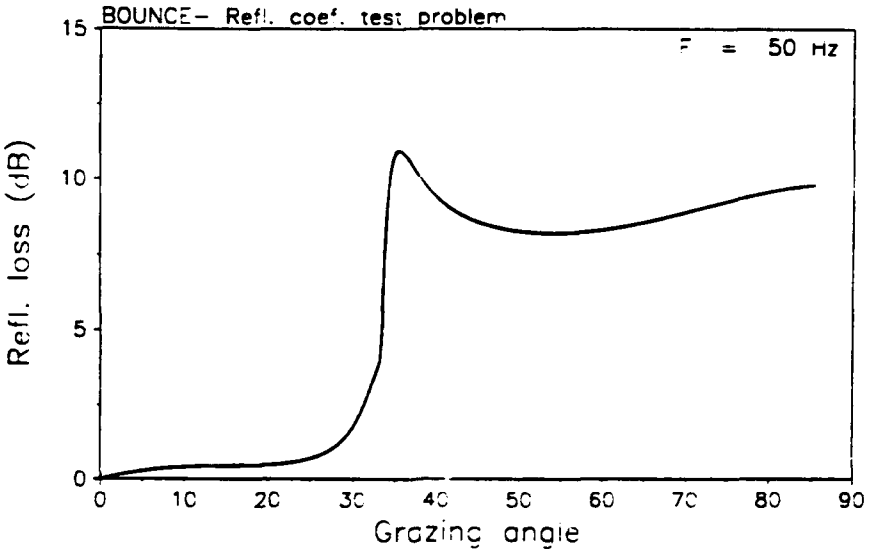


Figure 4.12: Sample output of PLOTRTH.

(3) - ANGLE AXIS INFO

Syntax:

THMIN THMAX NTH THINC THAXL

Description:

THMIN:	First angle	(degrees).
THMAX:	Last angle	(degrees).
THINC:	Angular interval for tick marks	(degrees).
THAXL:	Angle axis length	(cm).

(4) - REFLECTION LOSS AXIS INFO

Syntax:

RMIN RMAX RINC RAXL

Description:

RMIN:	R minimum	(dB).
RMAX:	R maximum	(dB).
RINC:	R interval for tick marks	(dB).
RAXL:	R axis length	(cm).

If you set RMIN=RMAX then the curve is autoscaled.

Chapter 5

Test Problems

The following test problems have been developed to validate the model by exercising various components of the code and to illustrate the input structure required for various kinds of scenarios. In brief, we have:

- PEKERIS: A simple (two-layer) Pekeris waveguide.
- TWERSKY: The Pekeris wave guide with surface roughness. Demonstrates that the Twersky scatter works properly.
- SCHOLTE: A two-layer waveguide with an elastic bottom which leads to a Scholte wave. Demonstrates that the elastic half-space condition functions correctly.
- DOUBLE: A double-duct problem demonstrating that gradients are handled properly.
- FLUSED: A three-layer problem involving ocean, sediment and half-space. Demonstrates that multiple layers are treated properly.
- ELSED: A three-layer problem with shear properties in the sediment. Demonstrates that elastic media are handled properly.
- ATTEN: A two-layer problem with volume attenuation. Demonstrates that attenuation is handled properly.
- NORMAL: A problem with several density changes to check out the modal normalization in a severe case.
- ICE: A problem with an elastic ice layer to demonstrate that elastic layers above the water column are handled properly.

For each of these cases, we provide the environmental file along with the print-out from KRAKEN. The CPU times printed were obtained on a 0.5 megaflop workstation.

In all cases, the frequency is chosen as 10 Hz and the transmission loss is computed for a source/receiver depth combination of 500 m and 2500 m respectively. The

transmission loss plots show an overlay of KRAKEN (solid line), KRAKENC (dotted line) and SCOOTER (dashed line) results. These results have also been checked against the NRL FSTFLD code which agrees to within 1 dB (usually less).

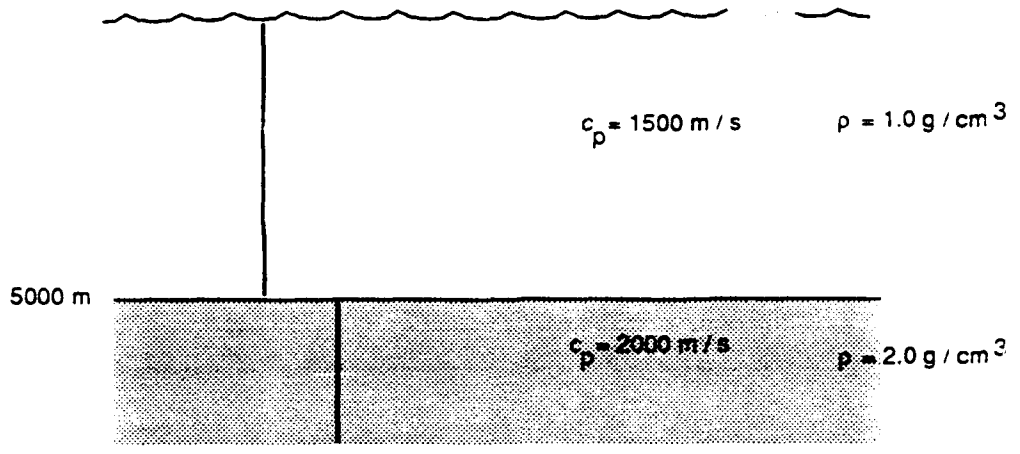


Figure 5.1: Schematic of the PEKERIS problem.

5.1 PEKERIS

This problem involves a homogeneous fluid layer with a sound speed of 1500 m/s overlying a faster bottom with sound speed 2000 m/s and density of 2.0 g/cm^3.

```
'Pekeris problem'
10.0
1
'NVF'
500  0.0  2
     0.0  1500.0 /
  5000.0  1500.0 /
'A'  0.0
  5000.0  2000.0  0.0  2.0 /
1400.0  2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

```
KRAKEN- Pekeris problem
Frequency =    10.00      NMEDIA =    1


    N2-LINEAR approximation to SSP
    Attenuation units: dB/mkHz
    VACUUM


     Z          ALPHAR      BETAR       RHO       ALPHAI      BETAI


         ( Number of pts =   500  RMS roughness =  0.000E+00 )
     0.00       1500.00       0.00      1.00       0.0000     0.0000
  5000.00       1500.00       0.00      1.00       0.0000     0.0000

                                    ( RMS roughness =  0.000E+00 )
    ACOUSTO-ELASTIC half-space
  5000.00       2000.00       0.00      2.00       0.0000     0.0000

CLOW =    1400.0       CHIGH =    2000.0
RMAX =     1000.000000000000

Number of sources   =             1
  500.0000

Number of receivers =             1
  2500.000

Mesh multiplier   CPU seconds
       1              5.49
       2              6.21

   I          K                  ALPHA          PHASE SPEED
   1   0.4188332253E-01   0.0000000000E+00    1500.164010
   2   0.4186958032E-01   0.0000000000E+00    1500.656385
   3   0.4184666447E-01   0.0000000000E+00    1501.478167
   4   0.4181455674E-01   0.0000000000E+00    1502.631092
   5   0.4177323161E-01   0.0000000000E+00    1504.117605
   6   0.4172265636E-01   0.0000000000E+00    1505.940862
   7   0.4166279103E-01   0.0000000000E+00    1508.104751
   8   0.4159358848E-01   0.0000000000E+00    1510.613904
   9   0.4151499439E-01   0.0000000000E+00    1513.473722
  10   0.4142694720E-01   0.0000000000E+00    1516.690399
  11   0.4132937809E-01   0.0000000000E+00    1520.270954
```

12	0.4122221089E-01	0.0000000000E+00	1524.223270
13	0.4110536194E-01	0.0000000000E+00	1528.556132
14	0.4097873993E-01	0.0000000000E+00	1533.279285
15	0.4084224568E-01	0.0000000000E+00	1538.403485
16	0.4069577186E-01	0.0000000000E+00	1543.940567
17	0.4053920272E-01	0.0000000000E+00	1549.903522
18	0.4037241363E-01	0.0000000000E+00	1556.306582
19	0.4019527072E-01	0.0000000000E+00	1563.165317
20	0.4000763035E-01	0.0000000000E+00	1570.496741
21	0.3980933859E-01	0.0000000000E+00	1578.319442
22	0.3960023053E-01	0.0000000000E+00	1586.653720
23	0.3938012967E-01	0.0000000000E+00	1595.521741
24	0.3914884708E-01	0.0000000000E+00	1604.947725
25	0.3890618058E-01	0.0000000000E+00	1614.958141
26	0.3865191380E-01	0.0000000000E+00	1625.581942
27	0.3838581509E-01	0.0000000000E+00	1636.850824
28	0.3810763645E-01	0.0000000000E+00	1648.799530
29	0.3781711221E-01	0.0000000000E+00	1661.466183
30	0.3751395766E-01	0.0000000000E+00	1674.892680
31	0.3719786754E-01	0.0000000000E+00	1689.125136
32	0.3686851438E-01	0.0000000000E+00	1704.214399
33	0.3652554677E-01	0.0000000000E+00	1720.216633
34	0.3616858743E-01	0.0000000000E+00	1737.194000
35	0.3579723130E-01	0.0000000000E+00	1755.215440
36	0.3541104368E-01	0.0000000000E+00	1774.357560
37	0.3500955866E-01	0.0000000000E+00	1794.705659
38	0.3459227830E-01	0.0000000000E+00	1816.354868
39	0.3415867360E-01	0.0000000000E+00	1839.411384
40	0.3370818983E-01	0.0000000000E+00	1863.993688
41	0.3324026217E-01	0.0000000000E+00	1890.233379
42	0.3275436107E-01	0.0000000000E+00	1918.274423
43	0.3225014368E-01	0.0000000000E+00	1948.265834
44	0.3172824619E-01	0.0000000000E+00	1980.312832

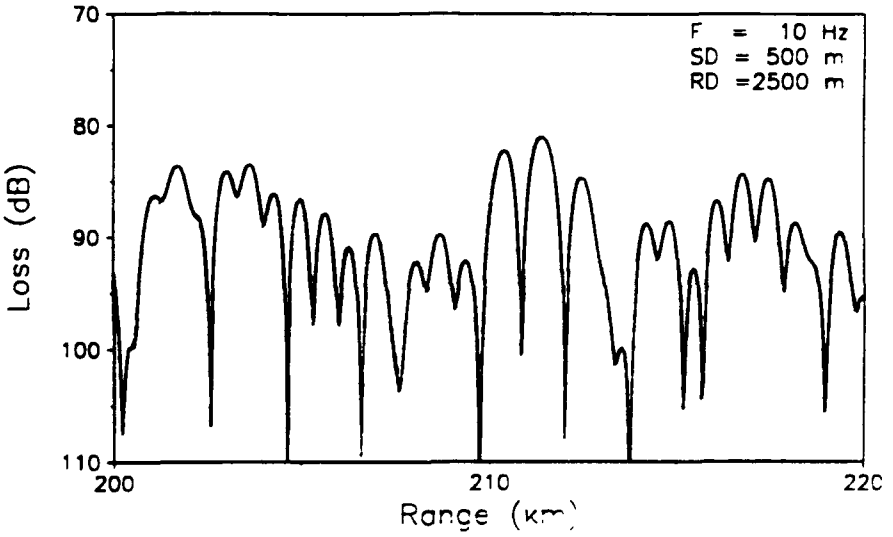


Figure 5.2: Transmission loss for the PEKERIS problem.

5.2 TWERSKY

The previous Pekeris problem is modified by the inclusion of surface scatter. The rough surface involves a density of 0.092 bosses per km of width 8.2 m and height 5.1 m. Note that the KRAKEN result differs from the KRAKENC and SCOOTER results. This reflects the error in using a perturbation theory which however is probably negligible considering the approximations of the scatter model.

```
'Pekeris problem with Twersky ice scatter'
10.0
1
'NSF'
0.092, 8.2, 5.1
500, 0.0, 2
    0.0, 1500.0 /
 5000.0, 1500.0 /
'A', 0.0
 5000.0, 2000.0,   0.0, 2.0, 0.0, 0.0
1400.0, 2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

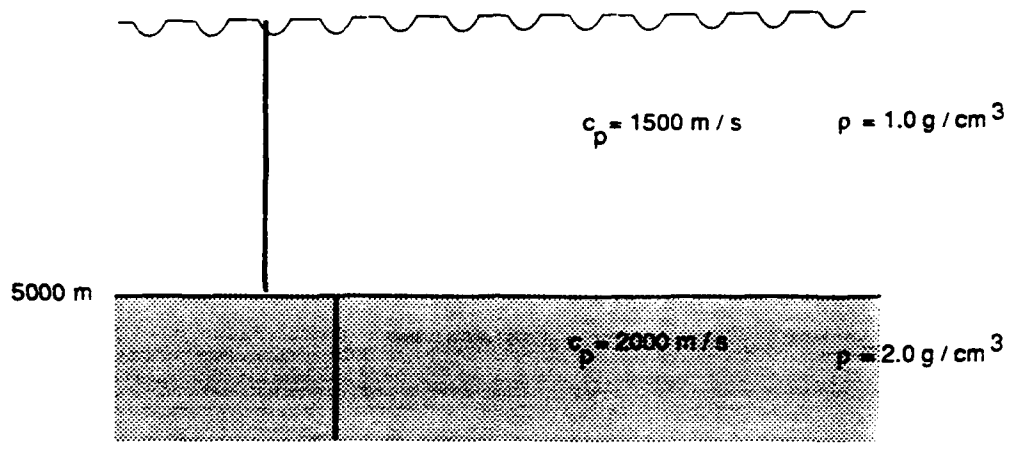


Figure 5.3: Schematic of the TWERSKY problem.

```
KRAKEN- Pekeris problem with Twersky ice scatter
Frequency =   10.00     NMEDIA =   1


    N2-LINEAR approximation to SSP
    Attenuation units: dB/mkHz
    Twersky SOFT BOSS scatter model

Twersky ice model parameters:
Bumden =    0.920000E-01  Eta =    8.20      Xi =    5.10


     Z          ALPHAR      BETAR       RHO        ALPHAI      BETAI


         ( Number of pts =   500  RMS roughness =  0.000E+00 )
     0.00       1500.00       0.00      1.00        0.0000     0.0000
  5000.00       1500.00       0.00      1.00        0.0000     0.0000

                                    ( RMS roughness =  0.000E+00 )
    ACOUSTO-ELASTIC half-space
  5000.00       2000.00       0.00      2.00        0.0000     0.0000

CLOW =   1400.0       CHIGH =   2000.0
RMAX =    1000.000000000000
```

```
Number of sources    =          1
  500.0000

Number of receivers =          1
  2500.000

Mesh multiplier   CPU seconds
       1             8.08
       2             6.19
```

I	K	ALPHA	PHASE SPEED
1	0.4188333967E-01	-0.7143639068E-09	1500.163396
2	0.4186964892E-01	-0.2858563396E-08	1500.653927
3	0.4184681891E-01	-0.6435953313E-08	1501.472626
4	0.4181483151E-01	-0.1145206277E-07	1502.621218
5	0.4177366141E-01	-0.1791463612E-07	1504.102129
6	0.4172327607E-01	-0.2583358102E-07	1505.918494
7	0.4166363582E-01	-0.3522095285E-07	1508.074172
8	0.4159469382E-01	-0.4609093903E-07	1510.573761
9	0.4151639611E-01	-0.5845985229E-07	1513.422622
10	0.4142868154E-01	-0.7234613786E-07	1516.626905
11	0.4133148173E-01	-0.8777040005E-07	1520.193577
12	0.4122472102E-01	-0.1047554344E-06	1524.130462
13	0.4110831629E-01	-0.1233263186E-06	1528.446279
14	0.4098217683E-01	-0.1435104786E-06	1533.150699
15	0.4084620412E-01	-0.1653378169E-06	1538.254396
16	0.4070029155E-01	-0.1888408516E-06	1543.769115
17	0.4054432412E-01	-0.2140548873E-06	1549.707744
18	0.4037817804E-01	-0.2410182110E-06	1556.084403
19	0.4020172036E-01	-0.2697723168E-06	1562.914535
20	0.4001480842E-01	-0.3003621806E-06	1570.215017
21	0.3981728933E-01	-0.3328365176E-06	1578.004282
22	0.3960899938E-01	-0.3672481421E-06	1586.302458
23	0.3938976329E-01	-0.4036543176E-06	1595.131522
24	0.3915939350E-01	-0.4421171667E-06	1604.515480
25	0.3891768932E-01	-0.4827041259E-06	1614.480566
26	0.3866443598E-01	-0.5254884517E-06	1625.055467
27	0.3839940360E-01	-0.5705497741E-06	1636.271587
28	0.3812234607E-01	-0.6179747460E-06	1648.163336
29	0.3783299981E-01	-0.6678576894E-06	1660.768466
30	0.3753108238E-01	-0.7203013712E-06	1674.128458
31	0.3721629100E-01	-0.7754178193E-06	1688.288956
32	0.3688830091E-01	-0.8333291792E-06	1703.300275
33	0.3654676364E-01	-0.8941686039E-06	1719.217978
34	0.3619130513E-01	-0.9580810818E-06	1736.103543

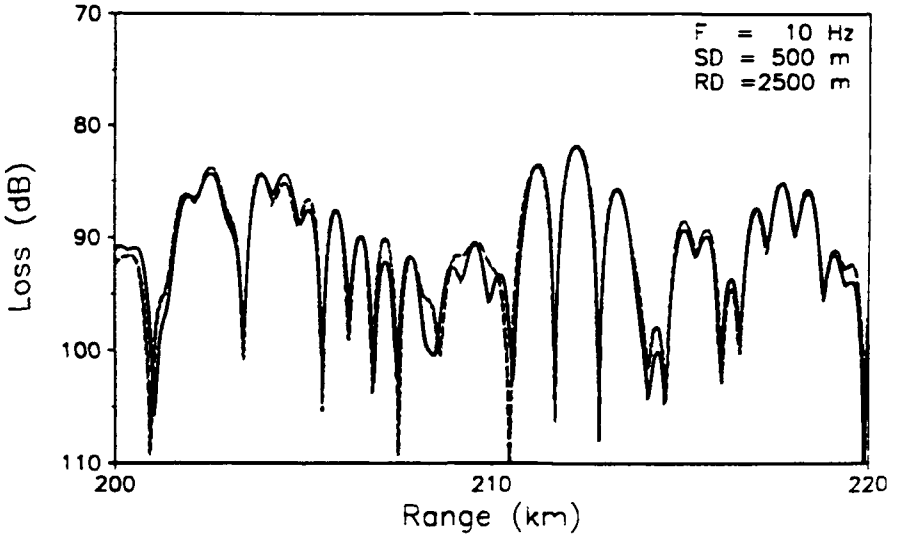


Figure 5.4: Transmission loss for the TWERSKY problem.

35	0.3582152384E-01	-0.1025224135E-05	1754.025132
36	0.3543698884E-01	-0.1095768032E-05	1773.058466
37	0.3503723827E-01	-0.1169895081E-05	1793.287833
38	0.3462177842E-01	-0.1247796785E-05	1814.807209
39	0.3419008460E-01	-0.1329666243E-05	1837.721486
40	0.3374160587E-01	-0.1415679057E-05	1862.147680
41	0.3327577965E-01	-0.1505943298E-05	1888.215805
42	0.3279207350E-01	-0.1600349816E-05	1916.068317
43	0.3229012280E-01	-0.1697989094E-05	1945.853643
44	0.3177042607E-01	-0.1793021126E-05	1977.683678

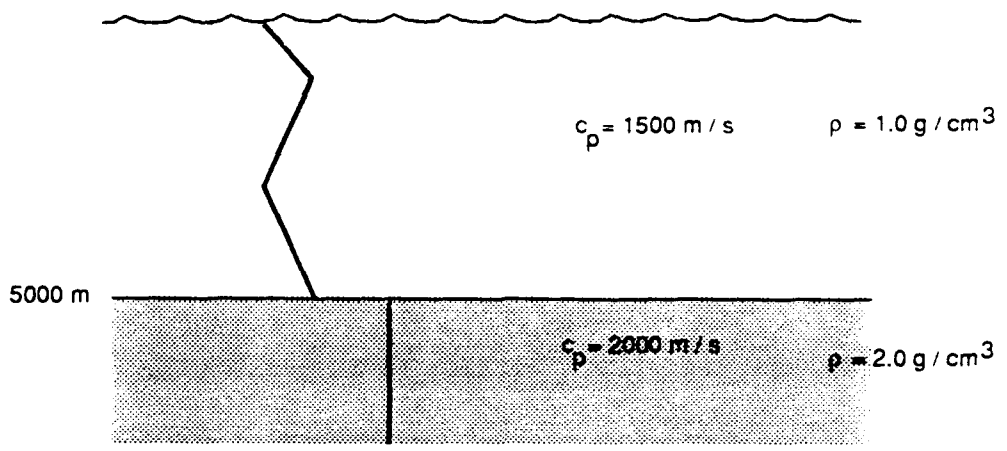


Figure 5.5: Schematic of the DOUBLE problem.

5.3 DOUBLE

The ocean profile is converted to one involving three piecewise linear segments defining a double-duct profile.

```
'Double-duct problem'
10.0
3
'NVF'
100 0.0 2
    0.0 1500.0 /
 1000.0 1550.0 /
200 0.0 2
 1000.0 1550.0 /
 3000.0 1500.0 /
200 0.0 2
 3000.0 1500.0 /
 5000.0 1550.0 /
'A' 0.0
 5000.0 2000.0   0.0 2.0 0.0 0.0
1400.0 2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

```
KRAKEN- Double-duct problem
Frequency =   10.00     NMEDIA =   3


    N2-LINEAR approximation to SSP
    Attenuation units: dB/mkHz
    VACUUM


     Z          ALPHAR      BETAR      RHO       ALPHAI      BETAI


         ( Number of pts =   100  RMS roughness =  0.000E+00 )
    0.00       1500.00       0.00      1.00       0.0000     0.0000
 1000.00       1550.00       0.00      1.00       0.0000     0.0000

         ( Number of pts =   200  RMS roughness =  0.000E+00 )
 1000.00       1550.00       0.00      1.00       0.0000     0.0000
 3000.00       1500.00       0.00      1.00       0.0000     0.0000

         ( Number of pts =   200  RMS roughness =  0.000E+00 )
 3000.00       1500.00       0.00      1.00       0.0000     0.0000
 5000.00       1550.00       0.00      1.00       0.0000     0.0000

                                  ( RMS roughness =  0.000E+00 )
    ACOUSTO-ELASTIC half-space
 5000.00       2000.00       0.00      2.00       0.0000     0.0000

CLOW =   1400.0      CHIGH =   2000.0
RMAX =    1000.000000000000

Number of sources   =            1
  500.0000

Number of receivers =            1
  2500.000

Mesh multiplier   CPU seconds
       1             5.60
       2             6.18

   I          K                  ALPHA          PHASE SPEED
   1   0.4171018652E-01   0.0000000000E+00     1506.391084
   2   0.4147891740E-01   0.0000000000E+00     1514.790091
   3   0.4131862874E-01   0.0000000000E+00     1520.666464
```

4	0.4123681174E-01	0.0000000000E+00	1523.683583
5	0.4117017415E-01	0.0000000000E+00	1526.149801
6	0.4104029641E-01	0.0000000000E+00	1530.979515
7	0.4091561041E-01	0.0000000000E+00	1535.645013
8	0.4080128302E-01	0.0000000000E+00	1539.947973
9	0.4074949725E-01	0.0000000000E+00	1541.904988
10	0.4068324597E-01	0.0000000000E+00	1544.415928
11	0.4057281144E-01	0.0000000000E+00	1548.619650
12	0.4046123964E-01	0.0000000000E+00	1552.889967
13	0.4035440690E-01	0.0000000000E+00	1557.001029
14	0.4024224926E-01	0.0000000000E+00	1561.340487
15	0.4011172669E-01	0.0000000000E+00	1566.421051
16	0.3996592323E-01	0.0000000000E+00	1572.135660
17	0.3980769235E-01	0.0000000000E+00	1578.384713
18	0.3964207800E-01	0.0000000000E+00	1584.978796
19	0.3946677171E-01	0.0000000000E+00	1592.019067
20	0.3927946746E-01	0.0000000000E+00	1599.610614
21	0.3907987820E-01	0.0000000000E+00	1607.780166
22	0.3886748929E-01	0.0000000000E+00	1616.565778
23	0.3864545686E-01	0.0000000000E+00	1625.853546
24	0.3841222010E-01	0.0000000000E+00	1635.725634
25	0.3816711818E-01	0.0000000000E+00	1646.229951
26	0.3790948500E-01	0.0000000000E+00	1657.417743
27	0.3763853318E-01	0.0000000000E+00	1669.349142
28	0.3735627690E-01	0.0000000000E+00	1681.962398
29	0.3706135033E-01	0.0000000000E+00	1695.347107
30	0.3675356291E-01	0.0000000000E+00	1709.544548
31	0.3643204686E-01	0.0000000000E+00	1724.631430
32	0.3609604877E-01	0.0000000000E+00	1740.685067
33	0.3574683553E-01	0.0000000000E+00	1757.689937
34	0.3538311960E-01	0.0000000000E+00	1775.757869
35	0.3500480248E-01	0.0000000000E+00	1794.949511
36	0.3461083089E-01	0.0000000000E+00	1815.381239
37	0.3420046728E-01	0.0000000000E+00	1837.163585
38	0.3377442369E-01	0.0000000000E+00	1860.338274
39	0.3333144286E-01	0.0000000000E+00	1885.062502
40	0.3287145204E-01	0.0000000000E+00	1911.441362
41	0.3239342265E-01	0.0000000000E+00	1939.648482
42	0.3189739326E-01	0.0000000000E+00	1969.811532

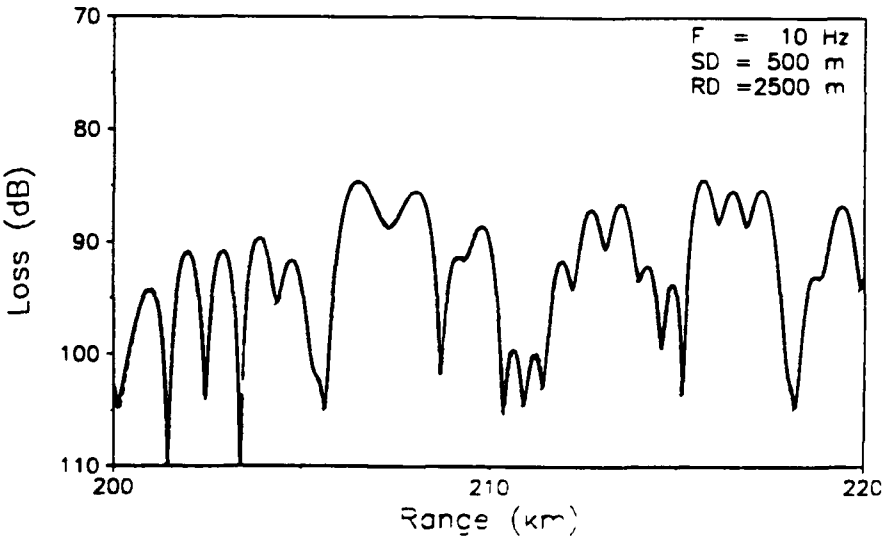


Figure 5.6: Transmission loss for the DOUBLE problem.

5.4 SCHOLTE

This problem is a version of the Pekeris waveguide but with an elastic half-space as the bottom. This type of problem has a Scholte mode with a phase velocity less than the slowest speed in the problem. (Since the source and receiver are many wavelenghts from the interface the Scholte mode is not actually important for the transmission loss calculation.)

```
'Scholte waveguide'
10.0
1
'NVM'
500 0.0 2
   0.0 1500.0 /
5000.0 1500.0 /
'A' 0.0
5000.0 4000.0 2000.0 2.0 /
1400.0 2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

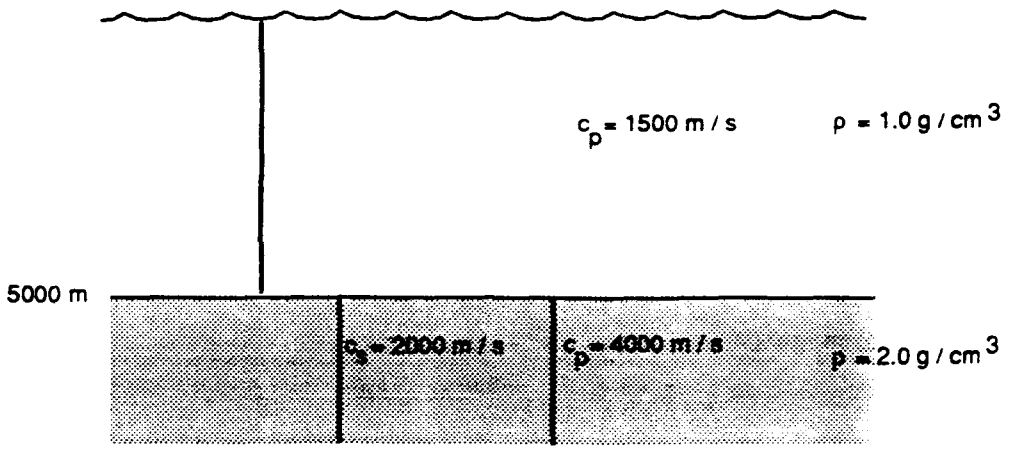


Figure 5.7: Schematic of the SCHOLTE problem.

```
KRAKEN- Scholte waveguide
Frequency =    10.00      NMEDIA =    1


    N2-LINEAR approximation to SSP
    Attenuation units: dB/m
    VACUUM


     Z           ALPHAR      BETAR       RHO        ALPHAI      BETAI


          ( Number of pts =    500  RMS roughness =  0.000E+00 )
      0.00       1500.00       0.00      1.00        0.0000     0.0000
   5000.00       1500.00       0.00      1.00        0.0000     0.0000

                                    ( RMS roughness =  0.000E+00 )
    ACOUSTO-ELASTIC half-space
   5000.00       4000.00    2000.00      2.00        0.0000     0.0000

CLOW =    1400.0       CHIGH =    2000.0
RMAX =     1000.000000000000

Number of sources   =             1
  500.0000

Number of receivers =             1
  2500.000
```

Mesh multiplier	CPU seconds
1	5.61
2	6.51
4	4.64

I	K	ALPHA	PHASE SPEED
1	0.4400982929E-01	0.0000000000E+00	1427.677728
2	0.4188306870E-01	0.0000000000E+00	1500.173101
3	0.4186856672E-01	0.0000000000E+00	1500.692715
4	0.4184439022E-01	0.0000000000E+00	1501.559773
5	0.4181052921E-01	0.0000000000E+00	1502.775838
6	0.4176696933E-01	0.0000000000E+00	1504.343123
7	0.4171369148E-01	0.0000000000E+00	1506.264510
8	0.4165067147E-01	0.0000000000E+00	1508.543581
9	0.4157787955E-01	0.0000000000E+00	1511.184643
10	0.4149527997E-01	0.0000000000E+00	1514.192774
11	0.4140283053E-01	0.0000000000E+00	1517.573853
12	0.4130048217E-01	0.0000000000E+00	1521.334613
13	0.4118817854E-01	0.0000000000E+00	1525.482682
14	0.4106585562E-01	0.0000000000E+00	1530.026639
15	0.4093344142E-01	0.0000000000E+00	1534.976071
16	0.4079085559E-01	0.0000000000E+00	1540.341632
17	0.4063800914E-01	0.0000000000E+00	1546.135118
18	0.4047480418E-01	0.0000000000E+00	1552.369538
19	0.4030113364E-01	0.0000000000E+00	1559.059197
20	0.4011688100E-01	0.0000000000E+00	1566.219793
21	0.3992192010E-01	0.0000000000E+00	1573.868514
22	0.3971611492E-01	0.0000000000E+00	1582.024153
23	0.3949931943E-01	0.0000000000E+00	1590.707232
24	0.3927137754E-01	0.0000000000E+00	1599.940135
25	0.3903212305E-01	0.0000000000E+00	1609.747258
26	0.3878137986E-01	0.0000000000E+00	1620.155170
27	0.3851896232E-01	0.0000000000E+00	1631.192776
28	0.3824467597E-01	0.0000000000E+00	1642.891500
29	0.3795831866E-01	0.0000000000E+00	1655.285463
30	0.3765968244E-01	0.0000000000E+00	1668.411654
31	0.3734855636E-01	0.0000000000E+00	1682.310086
32	0.3702473075E-01	0.0000000000E+00	1697.023903
33	0.3668800348E-01	0.0000000000E+00	1712.599409
34	0.3633818906E-01	0.0000000000E+00	1729.085975
35	0.3597513167E-01	0.0000000000E+00	1746.535736
36	0.3559872314E-01	0.0000000000E+00	1765.002998
37	0.3520892659E-01	0.0000000000E+00	1784.543272
38	0.3480580404E-01	0.0000000000E+00	1805.211941

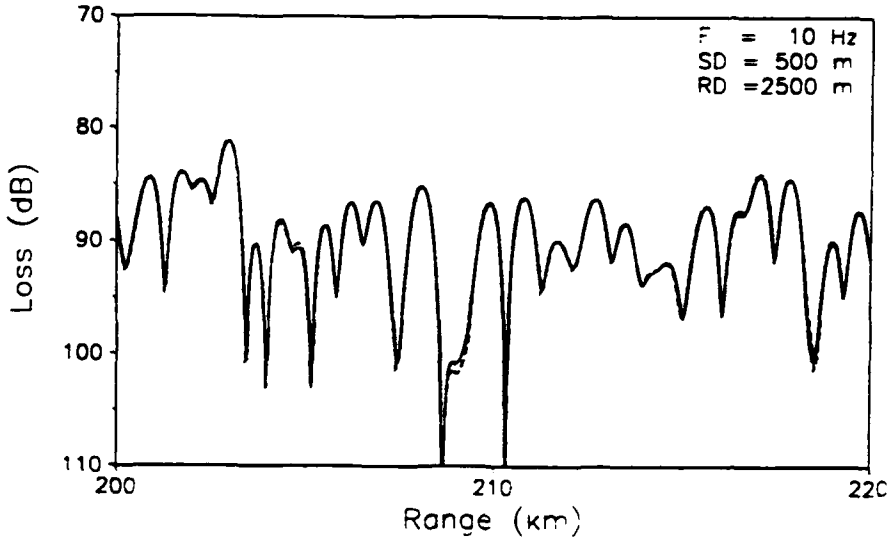


Figure 5.8: Transmission loss for the SCHOLTE problem.

39	0.3438954040E-01	0.0000000000E+00	1827.062890
40	0.3396044231E-01	0.0000000000E+00	1850.148255
41	0.3351886824E-01	0.0000000000E+00	1874.521915
42	0.3306503451E-01	0.0000000000E+00	1900.250643
43	0.3259872074E-01	0.0000000000E+00	1927.433091
44	0.3211925126E-01	0.0000000000E+00	1956.205410
45	0.3162787575E-01	0.0000000000E+00	1986.597316

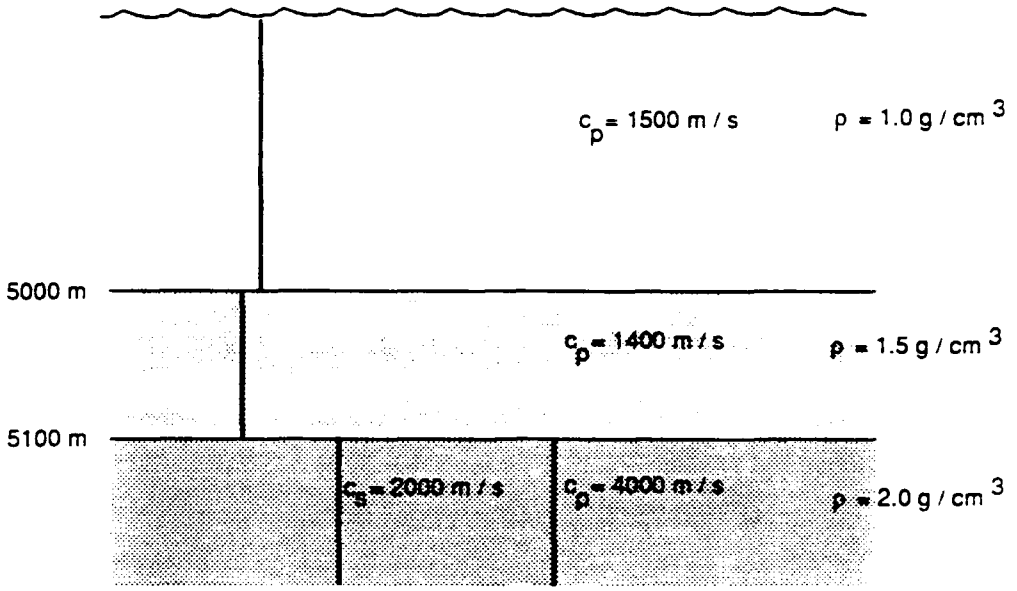


Figure 5.9: Schematic of the FLUSED problem.

5.5 FLUSED

A fluid sediment is inserted between the bottom half-space and the ocean.

```
'Fluid sediment problem'
10.0
2
'NVF'
500 0.0 2
    0.0 1500.0 /
 5000.0 1500.0 /
200 0.0 2
 5000.0 1400.0    0.0 1.5 /
 5100.0 1400.0    0.0 1.5 /
'A' 0.0
 5100.0 4000.0 2000.0 2.0 /
1300.0 2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

```
KRAKEN- Fluid sediment problem
Frequency =   10.00     NMEDIA =   2


    N2-LINEAR approximation to SSP
    Attenuation units: dB/mkHz
    VACUUM


     Z          ALPHAR     BETAR      RHO       ALPHAI     BETAI


         ( Number of pts =   500  RMS roughness =  0.000E+00 )
     0.00       1500.00      0.00     1.00       0.0000     0.0000
  5000.00       1500.00      0.00     1.00       0.0000     0.0000

         ( Number of pts =   200  RMS roughness =  0.000E+00 )
  5000.00       1400.00      0.00     1.50       0.0000     0.0000
  5100.00       1400.00      0.00     1.50       0.0000     0.0000

                                  ( RMS roughness =  0.000E+00 )
    ACOUSTO-ELASTIC half-space
  5100.00       4000.00   2000.00     2.00       0.0000     0.0000

CLOW =   1300.0      CHIGH =   2000.0
RMAX =    1000.000000000000

Number of sources   =           1
  500.0000

Number of receivers =           1
  2500.000

Mesh multiplier   CPU seconds
       1             7.96
       2             9.19

   I          K                ALPHA          PHASE SPEED
   1   0.4762029270E-01   0.0000000000E+00    1319.434416
   2   0.4188346068E-01   0.0000000000E+00    1500.159062
   3   0.4187012980E-01   0.0000000000E+00    1500.636692
   4   0.4184788935E-01   0.0000000000E+00    1501.434219
   5   0.4181670690E-01   0.0000000000E+00    1502.553829
   6   0.4177653892E-01   0.0000000000E+00    1503.998529
   7   0.4172733229E-01   0.0000000000E+00    1505.772107
```

8	0.4166902579E-01	0.0000000000E+00	1507.879099
9	0.4160155146E-01	0.0000000000E+00	1510.324756
10	0.4152483561E-01	0.0000000000E+00	1513.115035
11	0.4143879961E-01	0.0000000000E+00	1516.256592
12	0.4134336028E-01	0.0000000000E+00	1519.756804
13	0.4123843013E-01	0.0000000000E+00	1523.623787
14	0.4112391725E-01	0.0000000000E+00	1527.866441
15	0.4099972515E-01	0.0000000000E+00	1532.494495
16	0.4086575247E-01	0.0000000000E+00	1537.518565
17	0.4072189255E-01	0.0000000000E+00	1542.950220
18	0.4056803308E-01	0.0000000000E+00	1548.802057
19	0.4040405563E-01	0.0000000000E+00	1555.087778
20	0.4022983521E-01	0.0000000000E+00	1561.822283
21	0.4004523991E-01	0.0000000000E+00	1569.021767
22	0.3985013046E-01	0.0000000000E+00	1576.703824
23	0.3964435995E-01	0.0000000000E+00	1584.887564
24	0.3942777350E-01	0.0000000000E+00	1593.593741
25	0.3920020817E-01	0.0000000000E+00	1602.844883
26	0.3896149284E-01	0.0000000000E+00	1612.665442
27	0.3871144846E-01	0.0000000000E+00	1623.081945
28	0.3844988844E-01	0.0000000000E+00	1634.123157
29	0.3817661955E-01	0.0000000000E+00	1645.820238
30	0.3789144340E-01	0.0000000000E+00	1658.206904
31	0.3759415875E-01	0.0000000000E+00	1671.319565
32	0.3728456516E-01	0.0000000000E+00	1685.197422
33	0.3696246847E-01	0.0000000000E+00	1699.882494
34	0.3662768905E-01	0.0000000000E+00	1715.419528
35	0.3628007398E-01	0.0000000000E+00	1731.855704
36	0.3591951486E-01	0.0000000000E+00	1749.240025
37	0.3554597307E-01	0.0000000000E+00	1767.622255
38	0.3515951389E-01	0.0000000000E+00	1787.051245
39	0.3476034709E-01	0.0000000000E+00	1807.572661
40	0.3434886085E-01	0.0000000000E+00	1829.226691
41	0.3392560858E-01	0.0000000000E+00	1852.047928
42	0.3349116237E-01	0.0000000000E+00	1876.072630
43	0.3304572812E-01	0.0000000000E+00	1901.360831
44	0.3258859668E-01	0.0000000000E+00	1928.031872
45	0.3211808726E-01	0.0000000000E+00	1956.276305
46	0.3163418448E-01	0.0000000000E+00	1986.201133

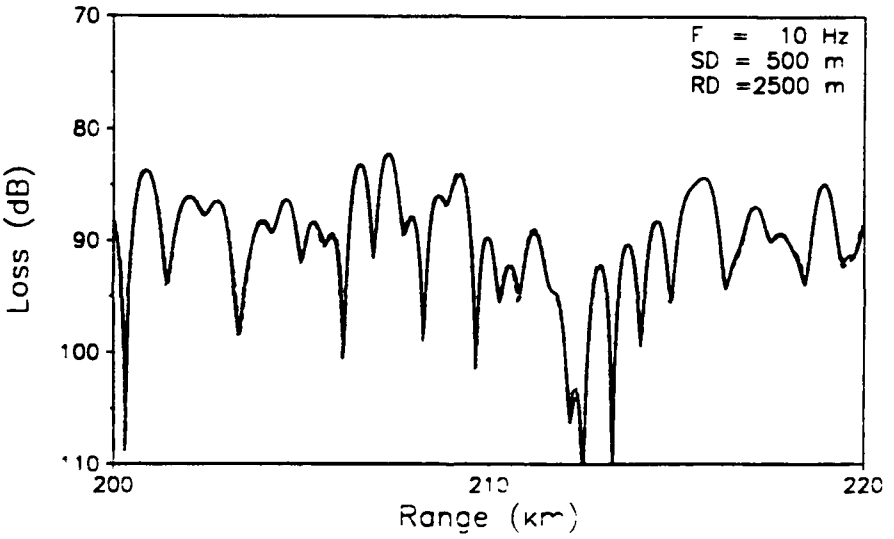


Figure 5.10: Transmission loss for the FLUSED problem.

5.6 ELSED

The previous problem (FLUSED) is modified by including shear properties in the sediment. This problem has several interfacial modes with phase velocities below 1300 m/s which have been excluded from the calculation.

```
'Elastic sediment problem'
10.0
2
'NVF'
500 0.0 2
    0.0 1500.0 /
 5000.0 1500.0 /
200 0.0 2
 5000.0 1400.0  700.0 1.5 /
 5100.0 1400.0  700.0 1.5 /
'A' 0.0
 5100.0 4000.0 2000.0 2.0 /
1300.0 2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

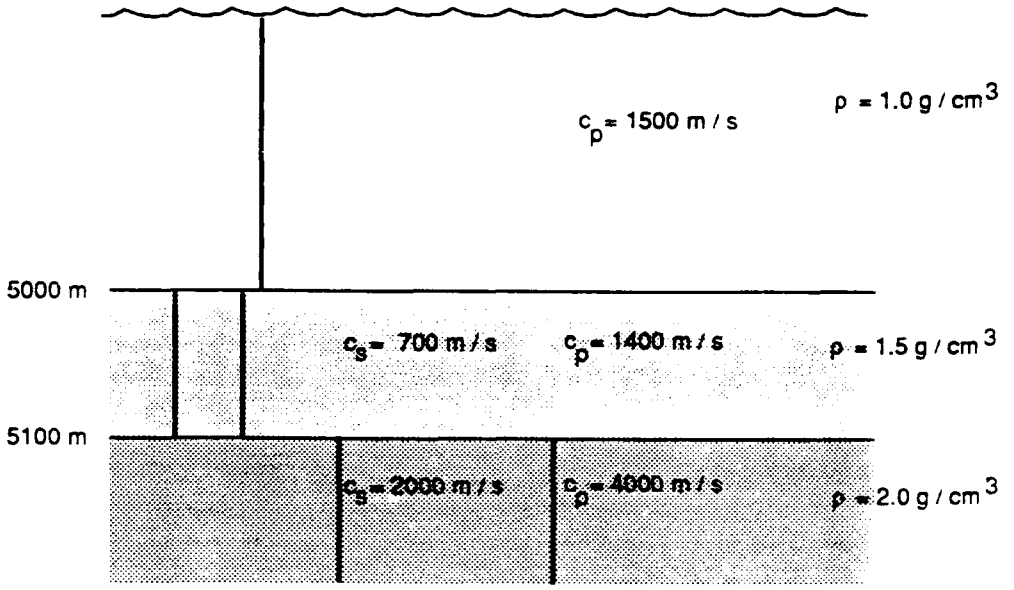


Figure 5.11: Schematic of the ELSED problem.

```
KRAKEN- Elastic sediment problem
Frequency =    10.00      NMEDIA =    2

    N2-LINEAR approximation to SSP
    Attenuation units: dB/mkHz
    VACUUM

     Z          ALPHAR      BETAR       RHO        ALPHAI      BETAI

          ( Number of pts =    500  RMS roughness =  0.000E+00 )
     0.00       1500.00       0.00      1.00        0.0000     0.0000
  5000.00       1500.00       0.00      1.00        0.0000     0.0000

          ( Number of pts =    200  RMS roughness =  0.000E+00 )
  5000.00       1400.00     700.00      1.50        0.0000     0.0000
  5100.00       1400.00     700.00      1.50        0.0000     0.0000

                                   ( RMS roughness =  0.000E+00 )
    ACOUSTO-ELASTIC half-space
  5100.00       4000.00    2000.00      2.00        0.0000     0.0000
```

```
CLOW =   1300.0      CHIGH =   2000.0
RMAX =    1000.000000000000

Number of sources   =           1
  500.0000

Number of receivers =           1
  2500.000

Mesh multiplier   CPU seconds
       1             23.9
       2             38.2
```

I	K	ALPHA	PHASE SPEED
1	0.4271788618E-01	0.0000000000E+00	1470.855857
2	0.4188323798E-01	0.0000000000E+00	1500.167038
3	0.4186924441E-01	0.0000000000E+00	1500.668425
4	0.4184591718E-01	0.0000000000E+00	1501.504981
5	0.4181324891E-01	0.0000000000E+00	1502.678091
6	0.4177122866E-01	0.0000000000E+00	1504.189728
7	0.4171984138E-01	0.0000000000E+00	1506.042473
8	0.4165906753E-01	0.0000000000E+00	1508.239545
9	0.4158888289E-01	0.0000000000E+00	1510.784823
10	0.4150925843E-01	0.0000000000E+00	1513.682861
11	0.4142016032E-01	0.0000000000E+00	1516.938916
12	0.4132155011E-01	0.0000000000E+00	1520.558955
13	0.4121338489E-01	0.0000000000E+00	1524.549688
14	0.4109561752E-01	0.0000000000E+00	1528.918577
15	0.4096819680E-01	0.0000000000E+00	1533.673873
16	0.4083106766E-01	0.0000000000E+00	1538.824642
17	0.4068417123E-01	0.0000000000E+00	1544.380804
18	0.4052744476E-01	0.0000000000E+00	1550.353185
19	0.4036082144E-01	0.0000000000E+00	1556.753575
20	0.4018422979E-01	0.0000000000E+00	1563.594808
21	0.3999759286E-01	0.0000000000E+00	1570.890861
22	0.3980082677E-01	0.0000000000E+00	1578.656982
23	0.3959383877E-01	0.0000000000E+00	1586.909858
24	0.3937652458E-01	0.0000000000E+00	1595.667818
25	0.3914876508E-01	0.0000000000E+00	1604.951087
26	0.3891042253E-01	0.0000000000E+00	1614.782081
27	0.3866133664E-01	0.0000000000E+00	1625.185742
28	0.3840132111E-01	0.0000000000E+00	1636.189883
29	0.3813016132E-01	0.0000000000E+00	1647.825524
30	0.3784761400E-01	0.0000000000E+00	1660.127190
31	0.3755340932E-01	0.0000000000E+00	1673.133125

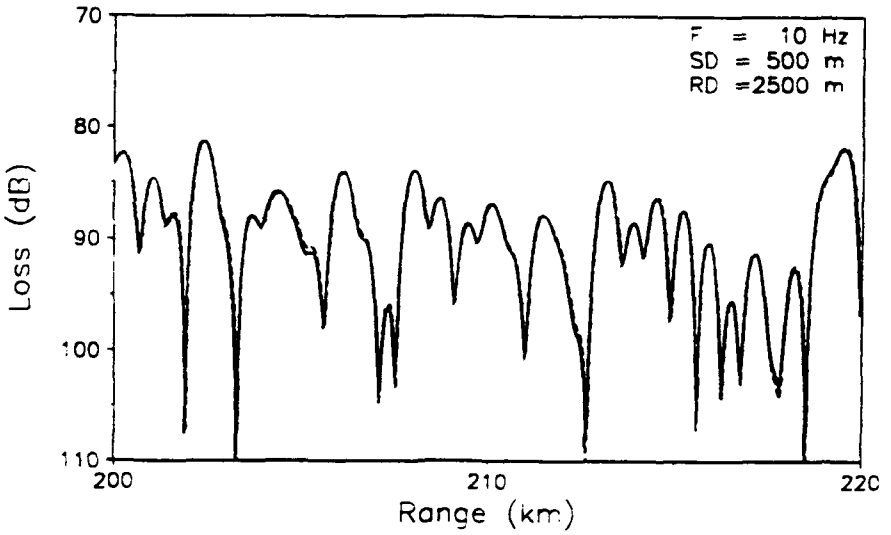


Figure 5.12: Transmission loss for the ELSED problem.

32	0.3724725578E-01	0.0000000000E+00	1686.885430
33	0.3692884747E-01	0.0000000000E+00	1701.430111
34	0.3659787314E-01	0.0000000000E+00	1716.817063
35	0.3625402627E-01	0.0000000000E+00	1733.100004
36	0.3589701555E-01	0.0000000000E+00	1750.336403
37	0.3552657515E-01	0.0000000000E+00	1768.587397
38	0.3514247440E-01	0.0000000000E+00	1787.917730
39	0.3474452577E-01	0.0000000000E+00	1808.395759
40	0.3433258869E-01	0.0000000000E+00	1830.093665
41	0.3390656461E-01	0.0000000000E+00	1853.088150
42	0.3346637756E-01	0.0000000000E+00	1877.462028
43	0.3301194133E-01	0.0000000000E+00	1903.306820
44	0.3254315545E-01	0.0000000000E+00	1930.724056
45	0.3206016806E-01	0.0000000000E+00	1959.810471
46	0.3156608684E-01	0.0000000000E+00	1990.485973

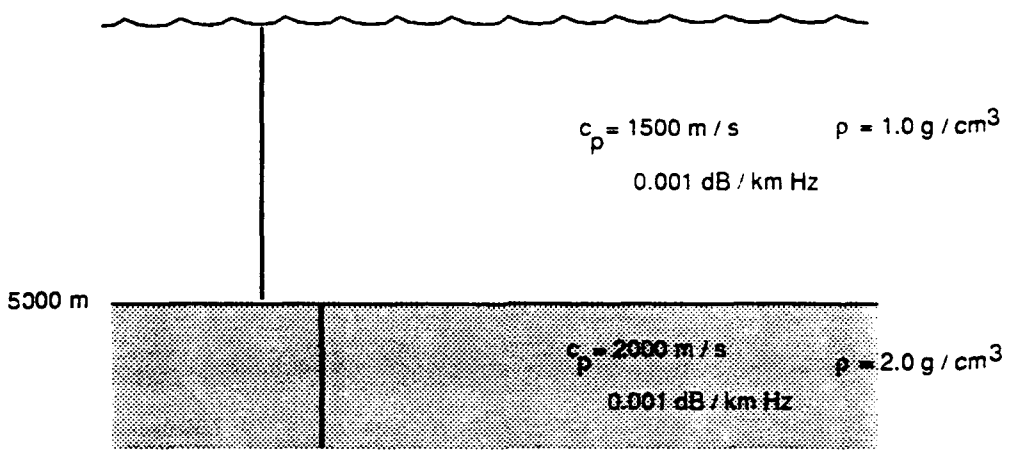


Figure 5.13: Schematic of the ATTEN problem.

5.7 ATTEN

Volume attenuation is included in both ocean and half-space.

```
'Attenuation test .001dB/kmHz'
10.0
1
'NVF'
500 0.0 2
    0.0 1500.0    0.0 1.0 0.001 0.0
 5000.0 1500.0    0.0 1.0 0.001 0.0
'A' 0.0
 5000.0 2000.0    0.0 2.0 0.001 0.0
1400.0 2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

```
KRAKEN- Attenuation test .001dB/kmHz
Frequency =   10.00     NMEDIA =   1


    N2-LINEAR approximation to SSP
    Attenuation units: dB/mkHz
    VACUUM


     Z          ALPHAR      BETAR       RHO       ALPHAI      BETAI


         ( Number of pts =   500  RMS roughness =  0.000E+00 )
    0.00       1500.00       0.00      1.00       0.0010     0.0000
 5000.00       1500.00       0.00      1.00       0.0010     0.0000

                                  ( RMS roughness =  0.000E+00 )
    ACOUSTO-ELASTIC half-space
 5000.00       2000.00       0.00      2.00       0.0010     0.0000

CLOW =   1400.0      CHIGH =   2000.0
RMAX =    1000.000000000000

Number of sources   =           1
  500.0000

Number of receivers =           1
  2500.000

Mesh multiplier   CPU seconds
       1             5.40
       2             6.08

   I          K                  ALPHA          PHASE SPEED
   1   0.4188332250E-01  -0.1151416386E-05    1500.164011
   2   0.4186958029E-01  -0.1151788191E-05    1500.656387
   3   0.4184666444E-01  -0.1152408844E-05    1501.478168
   4   0.4181455671E-01  -0.1153279807E-05    1502.631094
   5   0.4177323158E-01  -0.1154403131E-05    1504.117606
   6   0.4172265632E-01  -0.1155781449E-05    1505.940863
   7   0.4166279100E-01  -0.1157417981E-05    1508.104752
   8   0.4159358845E-01  -0.1159316538E-05    1510.613905
   9   0.4151499436E-01  -0.1161481525E-05    1513.473723
  10   0.4142694717E-01  -0.1163917959E-05    1516.690400
  11   0.4132937806E-01  -0.1166631476E-05    1520.270955
```

12	0.4122221086E-01	-0.1169628360E-05	1524.223271
13	0.4110536191E-01	-0.1172915566E-05	1528.556134
14	0.4097873990E-01	-0.1176500754E-05	1533.279287
15	0.4084224564E-01	-0.1180392329E-05	1538.403486
16	0.4069577183E-01	-0.1184599493E-05	1543.940568
17	0.4053920269E-01	-0.1189132294E-05	1549.903523
18	0.4037241360E-01	-0.1194001698E-05	1556.306584
19	0.4019527068E-01	-0.1199219655E-05	1563.165318
20	0.4000763032E-01	-0.1204799188E-05	1570.496742
21	0.3980933855E-01	-0.1210754481E-05	1578.319443
22	0.3960023050E-01	-0.1217100991E-05	1586.653721
23	0.3938012964E-01	-0.1223855559E-05	1595.521743
24	0.3914884705E-01	-0.1231036550E-05	1604.947727
25	0.3890618055E-01	-0.1238663998E-05	1614.958142
26	0.3865191376E-01	-0.1246759776E-05	1625.581943
27	0.3838581506E-01	-0.1255347785E-05	1636.850826
28	0.3810763642E-01	-0.1264454167E-05	1648.799531
29	0.3781711218E-01	-0.1274107542E-05	1661.466184
30	0.3751395763E-01	-0.1284339274E-05	1674.892681
31	0.3719786751E-01	-0.1295183771E-05	1689.125138
32	0.3686851435E-01	-0.1306678812E-05	1704.214400
33	0.3652554674E-01	-0.1318865902E-05	1720.216634
34	0.3616858740E-01	-0.1331790655E-05	1737.194002
35	0.3579723127E-01	-0.1345503170E-05	1755.215441
36	0.3541104365E-01	-0.1360058366E-05	1774.357562
37	0.3500955864E-01	-0.1375516165E-05	1794.705661
38	0.3459227827E-01	-0.1391941313E-05	1816.354869
39	0.3415867357E-01	-0.1409402284E-05	1839.411385
40	0.3370818980E-01	-0.1427967936E-05	1863.993689
41	0.3324026214E-01	-0.1447697921E-05	1890.233381
42	0.3275436104E-01	-0.1468612832E-05	1918.274424
43	0.3225014365E-01	-0.1490575913E-05	1948.265835
44	0.3172824616E-01	-0.1512451223E-05	1980.312834

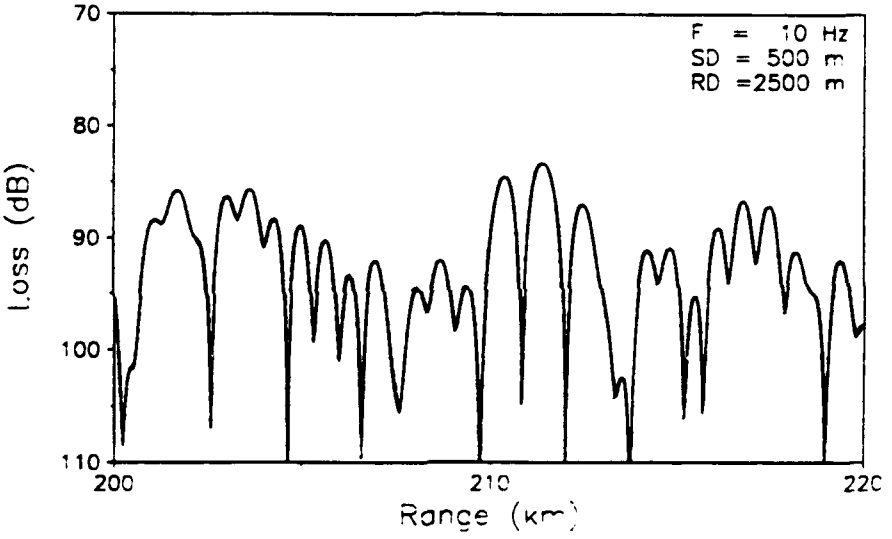


Figure 5.14: Transmission loss for the ATTEN problem.

5.8 NORMAL

Mode normalization is checked using several density changes. Due to the shear in the lower halfspace, there is a Scholte wave with a phase velocity of about 1393 m/s. It has been excluded from the calculation.

```
'Mode normalization test'
10.0
2
'NVF'
300 0.0 2
    0.0 1500.0 /
 3000.0 1500.0 /
200 0.0 2
 3000.0 1500.0    0.0 2.0 /
 5000.0 1500.0    0.0 2.0 /
'A' 0.0
 5000.0 4000.0 2000.0 3.0 /
1400.0 2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

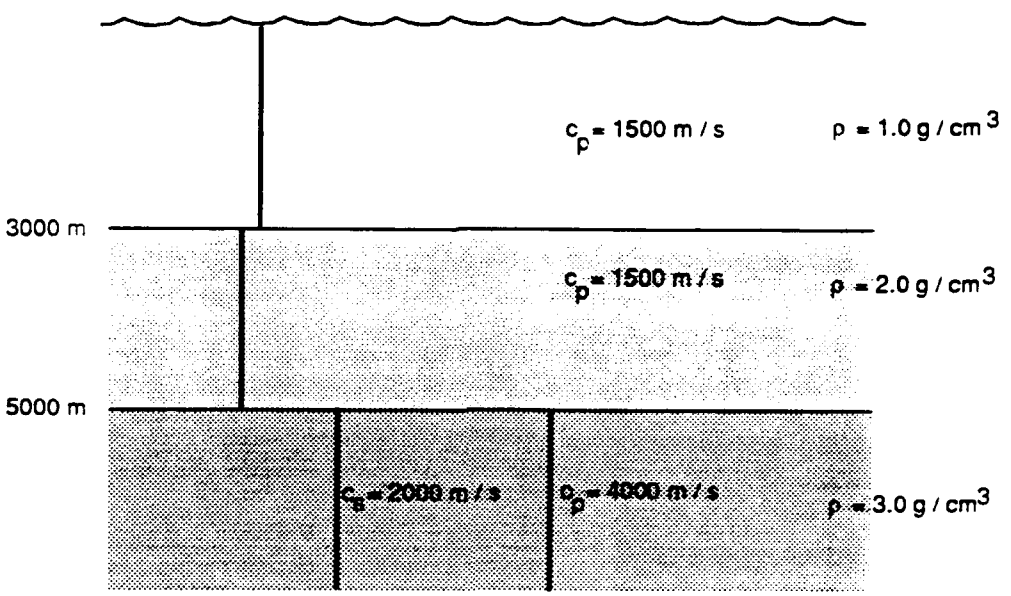


Figure 5.15: Schematic of the NORMAL problem.

```
KRAKEN- Mode normalization test
Frequency =   10.00     NMEDIA =   2

   N2-LINEAR approximation to SSP
   Attenuation units: dB/mkHz
   VACUUM

    Z          ALPHAR      BETAR      RHO        ALPHAI      BETAI

        ( Number of pts =   300  RMS roughness =  0.000E+00 )
   0.00       1500.00       0.00      1.00        0.0000     0.0000
3000.00       1500.00       0.00      1.00        0.0000     0.0000

        ( Number of pts =   200  RMS roughness =  0.000E+00 )
3000.00       1500.00       0.00      2.00        0.0000     0.0000
5000.00       1500.00       0.00      2.00        0.0000     0.0000

                                ( RMS roughness =  0.000E+00 )
  ACOUSTO-ELASTIC half-space
5000.00       4000.00    2000.00      3.00        0.0000     0.0000
```

```
CLOW =   1400.0      CHIGH =   2000.0
RMAX =    1000.000000000000

Number of sources   =            1
  500.0000

Number of receivers =            1
  2500.000

Mesh multiplier   CPU seconds
       1             5.55
       2             6.40
       4             4.54
```

I	K	ALPHA	PHASE SPEED
1	0.4188367900E-01	0.0000000000E+00	1500.151242
2	0.4186658699E-01	0.0000000000E+00	1500.763678
3	0.4184753177E-01	0.0000000000E+00	1501.447049
4	0.4180919493E-01	0.0000000000E+00	1502.823797
5	0.4176674150E-01	0.0000000000E+00	1504.351329
6	0.4171909765E-01	0.0000000000E+00	1506.069321
7	0.4164470397E-01	0.0000000000E+00	1508.759748
8	0.4158680917E-01	0.0000000000E+00	1510.860158
9	0.4149472381E-01	0.0000000000E+00	1514.213069
10	0.4140185763E-01	0.0000000000E+00	1517.609515
11	0.4131280307E-01	0.0000000000E+00	1520.880899
12	0.4117924681E-01	0.0000000000E+00	1525.813557
13	0.4108058576E-01	0.0000000000E+00	1529.478022
14	0.4093552018E-01	0.0000000000E+00	1534.898123
15	0.4078858493E-01	0.0000000000E+00	1540.427381
16	0.4065836697E-01	0.0000000000E+00	1545.360961
17	0.4046343001E-01	0.0000000000E+00	1552.805905
18	0.4032142497E-01	0.0000000000E+00	1558.274617
19	0.4012239980E-01	0.0000000000E+00	1566.004361
20	0.3991798210E-01	0.0000000000E+00	1574.023780
21	0.3974491332E-01	0.0000000000E+00	1580.877849
22	0.3948540371E-01	0.0000000000E+00	1591.267840
23	0.3929733220E-01	0.0000000000E+00	1598.883424
24	0.3904083516E-01	0.0000000000E+00	1609.388037
25	0.3877585468E-01	0.0000000000E+00	1620.386026
26	0.3855610943E-01	0.0000000000E+00	1629.621194
27	0.3822733064E-01	0.0000000000E+00	1643.636948
28	0.3799085539E-01	0.0000000000E+00	1653.867817
29	0.3766977091E-01	0.0000000000E+00	1667.964831
30	0.3734233184E-01	0.0000000000E+00	1682.590507

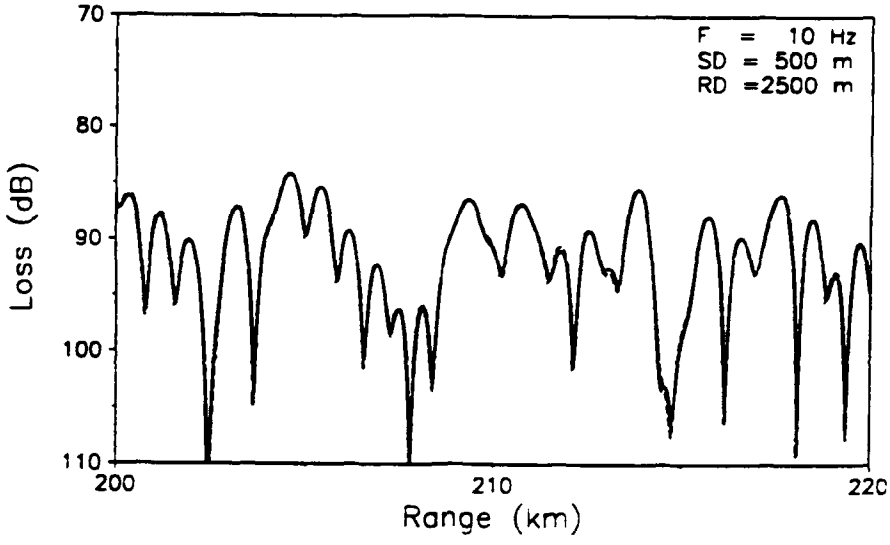


Figure 5.16: Transmission loss for the NORMAL problem.

31	0.3706900744E-01	0.0000000000E+00	1694.996910
32	0.3666482678E-01	0.0000000000E+00	1713.681983
33	0.3637907870E-01	0.0000000000E+00	1727.142504
34	0.3598056675E-01	0.0000000000E+00	1746.271911
35	0.3559408137E-01	0.0000000000E+00	1765.233169
36	0.3525354551E-01	0.0000000000E+00	1782.284652
37	0.3477030998E-01	0.0000000000E+00	1807.054729
38	0.3443753112E-01	0.0000000000E+00	1824.516771
39	0.3393983898E-01	0.0000000000E+00	1851.271395
40	0.3351944923E-01	0.0000000000E+00	1874.489424
41	0.3307617837E-01	0.0000000000E+00	1899.610419
42	0.3254372213E-01	0.0000000000E+00	1930.690436
43	0.3214824698E-01	0.0000000000E+00	1954.441034
44	0.3155272679E-01	0.0000000000E+00	1991.328784

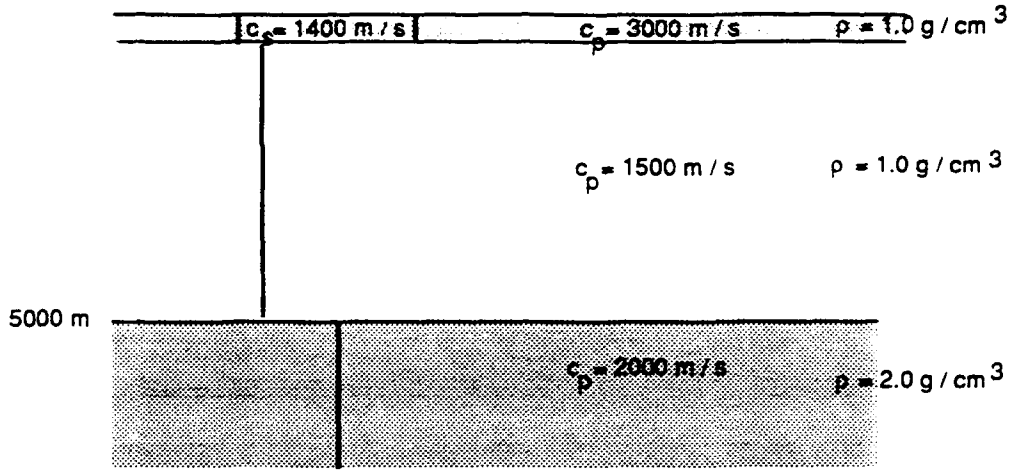


Figure 5.17: Schematic of the ICE problem.

5.9 ICE

This problem is loosely based on an Arctic scenario with an elastic ice-canopy. Here the elastic medium lies above the acoustic media. Note that the KRAKEN result disagrees with both KRAKENC and SCOOTER. This is expected since KRAKEN ignores attenuation in elastic media.

```
'Ice problem'
10.0
2
'NVW'
50 0.0 2
     0.0 3000.0 1400.0 1.0 0.3 1.0
    30.0 3000.0 1400.0 1.0 0.3 1.0
500 0.0 2
    30.0 1500.0    0.0 1.0 0.0 0.0
  5000.0 1500.0    0.0 1.0 0.0 0.0
'A' 0.0
  5000.0 2000.0    0.0 2.0 0.0 0.0
1400.0 2000.0
1000.0                    ! RMAX (km)
1  500.0 /                ! NSD  SD(1:NSD)
1 2500.0 /                ! NRD  RD(1:NRD)
```

```
KRAKEN- Ice problem
Frequency =   10.00     NMEDIA =   2


    N2-LINEAR approximation to SSP
    Attenuation units: dB/wavelength
    VACUUM


     Z          ALPHAR     BETAR      RHO       ALPHAI     BETAI


        ( Number of pts =    50  RMS roughness =  0.000E+00 )
     0.00       3000.00   1400.00     1.00       0.3000    1.0000
    30.00       3000.00   1400.00     1.00       0.3000    1.0000

        ( Number of pts =   500  RMS roughness =  0.000E+00 )
    30.00       1500.00      0.00     1.00       0.0000    0.0000
  5000.00       1500.00      0.00     1.00       0.0000    0.0000

                                  ( RMS roughness =  0.000E+00 )
    ACOUSTO-ELASTIC half-space
  5000.00       2000.00      0.00     2.00       0.0000    0.0000

CLOW =   1400.0      CHIGH =   2000.0
RMAX =    1000.000000000000

Number of sources   =           1
  500.0000

Number of receivers =           1
  2500.000

Mesh multiplier   CPU seconds
       1             11.5
       2             16.6

   I          K                ALPHA          PHASE SPEED
   1   0.4188333139E-01   0.0000000000E+00    1500.163692
   2   0.4186961576E-01   0.0000000000E+00    1500.655115
   3   0.4184674417E-01   0.0000000000E+00    1501.475307
   4   0.4181469833E-01   0.0000000000E+00    1502.626004
   5   0.4177345263E-01   0.0000000000E+00    1504.109646
   6   0.4172297425E-01   0.0000000000E+00    1505.929388
   7   0 4166322309E-01   0.0000000000E+00    1508.089111
```

8	0.4159415182E-01	0.0000000000E+00	1510.593445
9	0.4151570588E-01	0.0000000000E+00	1513.447784
10	0.4142782343E-01	0.0000000000E+00	1516.658320
11	0.4133043533E-01	0.0000000000E+00	1520.232066
12	0.4122346498E-01	0.0000000000E+00	1524.176900
13	0.4110682824E-01	0.0000000000E+00	1528.501608
14	0.4098043325E-01	0.0000000000E+00	1533.215930
15	0.4084418017E-01	0.0000000000E+00	1538.330622
16	0.4069796094E-01	0.0000000000E+00	1543.857521
17	0.4054165894E-01	0.0000000000E+00	1549.809621
18	0.4037514862E-01	0.0000000000E+00	1556.201159
19	0.4019829503E-01	0.0000000000E+00	1563.047712
20	0.4001095339E-01	0.0000000000E+00	1570.366306
21	0.3981296844E-01	0.0000000000E+00	1578.175543
22	0.3960417391E-01	0.0000000000E+00	1586.495737
23	0.3938439178E-01	0.0000000000E+00	1595.349077
24	0.3915343149E-01	0.0000000000E+00	1604.759805
25	0.3891108916E-01	0.0000000000E+00	1614.754417
26	0.3865714660E-01	0.0000000000E+00	1625.361895
27	0.3839137032E-01	0.0000000000E+00	1636.613972
28	0.3811351037E-01	0.0000000000E+00	1648.545423
29	0.3782329915E-01	0.0000000000E+00	1661.194409
30	0.3752045003E-01	0.0000000000E+00	1674.602864
31	0.3720465592E-01	0.0000000000E+00	1688.816938
32	0.3687558766E-01	0.0000000000E+00	1703.887506
33	0.3653289237E-01	0.0000000000E+00	1719.870752
34	0.3617619171E-01	0.0000000000E+00	1736.828840
35	0.3580508009E-01	0.0000000000E+00	1754.830681
36	0.3541912307E-01	0.0000000000E+00	1773.952815
37	0.3501785613E-01	0.0000000000E+00	1794.280405
38	0.3460078429E-01	0.0000000000E+00	1815.908349
39	0.3416738384E-01	0.0000000000E+00	1838.942465
40	0.3371710848E-01	0.0000000000E+00	1863.500635
41	0.3324940638E-01	0.0000000000E+00	1889.713529
42	0.3276376704E-01	0.0000000000E+00	1917.723716
43	0.3225987273E-01	0.0000000000E+00	1947.678269
44	0.3173836536E-01	0.0000000000E+00	1979.681448

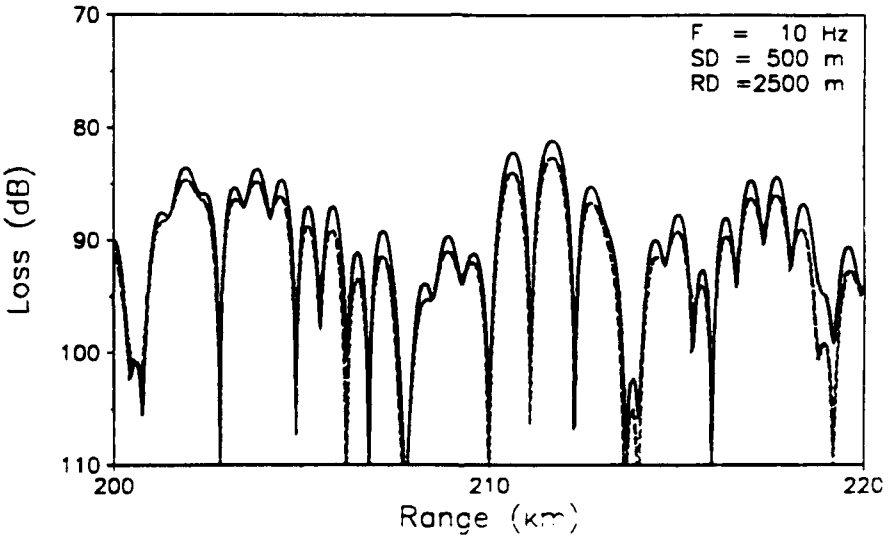


Figure 5.18: Transmission loss for the ICE problem.

Acknowledgments:

This work was partially supported by the Office of Naval Research Transients Program and the Office of Naval Technology High Gain Initiative Program.

Bibliography

[1] C. L. Pekeris, "Theory of propagation of explosive sound in shallow water," Geol. Soc. Amer. Mem. **27** (1948).

[2] A. O. Williams, "Normal-mode methods in propagation of underwater sound," in *Underwater Acoustics*, ed. R. W. B. Stephens, (Wiley-Interscience, New York, 1970).

[3] F. B. Jensen and M. C. Ferla, "SNAP: The SACLANTCEN normal-mode acoustic propagation model," SACLANTCEN Memorandum SM-121 (1979).

[4] D. F. Gordon, "Underwater sound propagation loss program," Naval Ocean Systems Center TR 393 (1979).

[5] B. A. Leverman, "User's guide to the normal mode propagation-loss package PROLOS," Defence Research Establishment Atlantic (informal communication) Research Note 82/3 (1982).

[6] D. D. Ellis, "A two-ended shooting technique for calculating normal modes in underwater acoustic propagation," Defence Research Establishment Atlantic report 85/105 (1985).

[7] C. L. Bartberger, "The computation of complex normal mode eigenvalues in underwater acoustic propagation," in *Computational Acoustics: Algorithms and Applications*, ed. D. Lee et al., Elsevier (1988).

[8] A. V. Newman and F. Ingenito, " A normal mode computer program for calculating sound propagation in shallow water with an arbitrary velocity profile," Naval Research Laboratory Memorandum Report 2381 (1972).

[9] H. M. Beisner, "Numerical calculation of normal modes for underwater sound propagation," IBM J. Res. Develop. **18**, 53–58 (1974).

[10] R. Gonzalez, "The numerical solution of the depth separated acoustic wave equation," Univ. of Texas at Austin, Master's thesis (1979).

[11] D. C. Stickler, "Normal-mode program with both the discrete and branch line contributions," J. Acoust. Soc. Am. **57**, 856–861 (1970).

[12] A. Nagl, G. L. Zarur and H. Überall, "A FORTRAN code for the calculation of sound propagation in a range-dependent ocean," Catholic Univ. report (1977).

[13] M. B. Porter, "A numerical method for computing ocean acoustic modes," Ph.D. dissertation, Northwestern Univ. (1984).

[14] M. B. Porter and E. L. Reiss, "A numerical method for ocean acoustic normal modes," J. Acoust. Soc. Am. **76**, 244–252 (1984).

[15] M. B. Porter and E. L. Reiss, "A numerical method for bottom interacting ocean acoustic normal modes," J. Acoust. Soc. Am. **77**, 1760–1767 (1985).

[16] M. B. Porter and E. L. Reiss, "A Numerical Method for Acoustic Normal Modes for Shear Flows", J. Sound and Vibration, **100**, 91–105 (1985).

[17] W. A. Kuperman, M. B. Porter, J. S. Perkins and R. B. Evans, "Rapid computation of acoustic fields in three-dimensional ocean environments," J. Acoust. Soc. Am. **89**, 125–133 (1991).

[18] J. S. Perkins, W.A. Kuperman, F. Ingenito and J. Glattetre, "Modeling ambient noise in three-dimensional ocean environments," J. Acoust. Soc. Am. in preparation, (1992).

[19] F. M. Labianca, "Normal modes, virtual modes, and alternative representations in the theory of surface-duct sound propagation," J. Acoust. Soc. Am. **53**, 1137–1147 (1973).

[20] S. T. McDaniel, "Mode coupling due to interaction with the seabed," J. Acoust. Soc. Am. **72**, 916–923 (1982).

[21] S. K. Mitchell and K. C. Focke, "The role of the seabottom attenuation profile in shallow water acoustic propagation," J. Acoust. Soc. Am. **73**, 465–473 (1983).

[22] S. R. Rutherford and K. E. Hawker, "An examination of the influence of the range dependence of the ocean bottom on the adiabatic approximation," J. Acoust. Soc. Am. **66**, 1145–1151 (1979).

[23] S. R. Rutherford and K. E. Hawker, "Effects of density gradients on bottom reflection loss for a class of marine sediments," J. Acoust. Soc. Am. **63**, 750–757 (1978).

[24] S. R. Rutherford, "An examination of multipath processes in a range dependent ocean environment within the context of adiabatic mode theory," J. Acoust. Soc. Am. **66**, 1482–1486 (1979).

[25] D. M. Milder, "Ray and wave invariants for SOFAR channel propagation," J. Acoust. Soc. Am. **46**, 1259–1263 (1969).

[26] M. C. Ferla, F. B. Jensen and W. A. Kuperman, "High-frequency mode calculations in deep water," J. Acoust. Soc. Am. **72**, 505–509 (1982).

[27] N. S. Ageeva and V. D. Krupin, "Structure of an infrasonic field in a shallow sea," Sov. Phys.-Acoustics **25**, 192–195 (1979).

[28] C. A. Boyles, *Acoustic Waveguides*, (Wiley, New York, 1984).

[29] C. A. Boyles, "Coupled mode solution for a cylindrically symmetric oceanic waveguide with range and depth dependent refractive index," J. Acoust. Soc. Am. **73**, 800–805 (1983).

[30] R. H. Ferris, "Comparison of measured and calculated normal-mode amplitude functions for acoustic waves in shallow water," J. Acoust. Soc. Am. **52**, 505–509 (1972).

[31] F. Ingenito, "Measurement of mode attenuation coefficients in shallow water," J. Acoust. Soc. Am. **53**, 858–863 (1973).

[32] C. T. Tindle, "Virtual modes and mode amplitudes near cutoff," J. Acoust. Soc. Am. **65**, 1423–1428 (1979).

[33] C. T. Tindle, K. M. Guthrie, G. E. J. Bold, M. D. Johns, K. O. Dixon and T. G. Birdsall, "Measurements of the frequency dependence of normal modes," J. Acoust. Soc. Am. **64**, 1178–1185 (1978).

[34] C. T. Tindle, "Attenuation parameters from normal mode measurements," J. Acoust. Soc. Am. **71**, 1145–1148 (1982).

[35] A. O. Williams, "Hidden depths: Acceptable ignorance about ocean bottoms," J. Acoust. Soc. Am. **59**, 1175–1179 (1976).

[36] H. R. Krol, "Some numerical considerations concerning eigenvalue problems in the theories of underwater sound and internal waves," SACLANTCEN Memorandum SM-40 (1974).

[37] L. B. Dozier and F. D. Tappert, "Statistics of normal mode amplitudes in a random ocean," J. Acoust. Soc. Am. **64**, 533–547 (1978).

[38] I. Tolstoy, "Guided waves in a fluid with continuously variable velocity overlying an elastic solid," J. Acoust. Soc. Am. **32**, 81–87 (1960).

[39] I. Tolstoy and J. May, "A numerical solution for the problem of long-range sound propagation in continuously stratified media, with applications to the deep ocean," J. Acoust. Soc. Am. **32**, 655–660 (1960).

[40] E. L. Ince, *Ordinary Differential Equations*, (Dover, New York, 1926).

[41] I. Stakgold, *Green's Functions and Boundary Value Problems*, (Wiley, New York, 1979).

[42] W. M. Ewing, W. S. Jardetzky and F. Press, *Elastic Waves in Layered Media*, (McGraw-Hill, New York, 1957).

[43] H. P. Bucker, "Sound propagation in a channel with lossy boundaries," J. Acoust. Soc. Am. **48**, 1187–1194 (1970).

[44] H. P. Bucker, "Wave propagation in a duct with boundary scattering (with application to surface duct)," J. Acoust. Soc. Am. **68**, 1768–1772 (1980).

[45] A. Haug, R. D. Graves and H. Überall, "Normal-mode theory of underwater sound propagation from directional multipole sources," J. Acoust. Soc. Am. **56**, 387–391 (1974).

[46] A. Nagl, H. Überall, A. J. Haug and G. L. Zarur, "Adiabatic mode theory of underwater sound propagation in a range-dependent environment," J. Acoust. Soc. Am. **63**, 739–749 (1978).

[47] W. H. Munk, "Sound channel in an exponentially stratifed ocean with applications to SOFAR," J. Acoust. Soc. Am. **55**, 220–226 (1974).

[48] C. M. Bender and S.A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers* (McGraw-Hill, New York, 1978).

[49] W. A. Kuperman and F. Ingenito, "Attenuation of the coherent component of sound propagating in shallow water with rough boundaries," J. Acoust. Soc. Am. **61**, 1178–1187 (1977).

[50] K. Aki and P. G. Richards, *Quantitative Seismology: Theory and Methods*, (Freeman, New York, 1980).

[51] O. I. Diachok, "Effects of sea-ice ridges on sound propagation in the Arctic Ocean," J. Acoust. Soc. Am. **59**, 1110–1120 (1976).

[52] S. C. Wales, "The NRL FSTFLD program: Operating instructions / User's manual," Naval Research Laboratory internal documents, (1986).

[53] R. J. Urick, *Principles of Underwater Sound*, (McGraw-Hill, New York, 1983).

[54] A. D. Pierce, "Extension of the method of normal modes to sound propagation in an almost-stratified medium," J. Acoust. Soc. Am. **37**, 19–27 (1965).

[55] H. Weinberg and R. Burridge, "Horizontal ray theory for ocean acoustics," J. Acoust. Soc. Am. **55**, 63–79 (1974).

[56] E. A. Lord and C. B. Wilson, *The Mathematical Desciption of Shape and Form*, (Wiley, New York, 1985).

[57] J. S. Perkins and R. N. Baer,"An approximation to the three-dimensional parabolic-equation method for acoustic propagation," J. Acoust. Soc. Am. **72**, 515–522 (1982).

[58] O. C. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, London, 1985.

[59] M. B. Porter and H. P. Bucker, "Gaussian beam tracing for computing ocean acoustic fields," J. Acoust. Soc. Am. **82**, 1349–1359 (1987).

[60] V. Červený, M. M. Popov and I. Pšenčík, "Computation of wave fields in inhomogeneous media— Gaussian beam approach," Geophys. J. R. astr. Soc. **70**,109–128 (1982).

[61] K. Yomogida, "Gaussian Beams for surface waves in laterally slowly-varying media," Geophys. J. R. astr. Soc. **82**, 511–533 (1985).

[62] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, (1965).

[63] R. P. Brent,"An algorithm with guaranteed convergence for finding a zero of a function," Comp. J. **14**, 422–425 (1971).

[64] H. B. Keller,"Accurate difference methods for linear ordinary differential systems subject to linear constraints," SIAM J. Numer. Anal., **6**, 8–30 (1969).

[65] R. B. Evans, "A coupled mode solution for acoustic propagation in a waveguide with stepwise depth variations of a penetrable bottom," J. Acoust. Soc. Am. **74**, 188–195 (1983).

[66] R. B. Evans and K. E. Gilbert, "The periodic extension of stepwise coupled modes," J. Acoust. Soc. Am. **77**, 983–988 (1985).

[67] M. B. Porter, F. B. Jensen and C. M. Ferla, "The problem of energy conservation in one-way models," J. Acoust. Soc. Am. **89**, 1058–1067 (1991).

[68] E. Ammicht and D. C. Stickler, "Uniform asymptotic evaluation of the continuous spectrum contribution for a stratified ocean," J. Acoust. Soc. Am. **76**, 186–191 (1984).

[69] R. A. Koch, P. J. Vidmar and J. B. Lindberg, "Normal mode identification for impedance boundary conditions," J. Acoust. Soc. Am. **73**, 1567–1570 (1983).

[70] R. A. Koch, C. Penland, P. J. Vidmar and K. E. Hawker, "On the calculation of normal mode group velocities and attenuation," J. Acoust. Soc. Am. **73**, 820–825 (1983).

[71] M. B. Porter and E. L. Reiss, "A Note on the Relationship Between Finite-Difference and Shooting Methods for ODE Eigenvalue Problems," SIAM J. Numerical Analysis **23**, 1034–1039 (1986).

[72] D. M. F. Chapman and D. D. Ellis, "The group velocity of normal modes," J. Acoust. Soc. Am. **74**, 973–979 (1983).

[73] M. A. Pedersen and D. F. Gordon, "Normal-mode and ray theory applied to underwater acoustic conditions of extreme downward refraction," J. Acoust. Soc. Am. **51**, 323–368 (1972).

[74] M. A. Pedersen and D. F. Gordon, "Normal-mode theory applied to short-range propagation in an underwater acoustic surface duct," J. Acoust. Soc. Am. **37**, 105–118 (1965).

[75] M. Hall, D. F. Gordon and D. White, "Improved methods for determining eigenfunctions in multi-layered normal-mode problems," J. Acoust. Soc. Am. **73**, 153–162 (1983).

[76] K. E. Hawker and T. L. Foreman, "A plane wave reflection loss model based on numerical integration," J. Acoust. Soc. Am. **64**, 1470–1477 (1978).

[77] M. D. Duston, G. R. Verma, D. H. Wood and R. P. Gilbert, "Direct generation of normal modes by transmutation theory," in *Computational Acoustics: Algorithms and Applications* (Proceedings of the 1st IMACS Symposium on Computational Acoustics, Yale), ed. D. Lee, R. L. Sternberg and M. H. Schultz, (North-Holland, pp. 389–402, 1990).

[78] R. F. Henrick and J. R. Brannan, "The uniform WKB modal approach to pulsed and broadband propagation," Johns Hopkins Univ.-Appl. Phys. Lab. STD-N-139 (1983).

[79] D. H. Wood, M. D. Duston and G. R. Verma, "Bottom interaction effects on normal modes: an algebraic approach," in *Ocean Seismo-Acoustics*, ed. T. Akal and J. Berkson (Plenum Press, New York, 1986).

[80] M. B. Porter, "The time-marched FFP for modeling acoustic pulse propagation", J. Acoust. Soc. Am. **87**, 2013–2023 (1990).